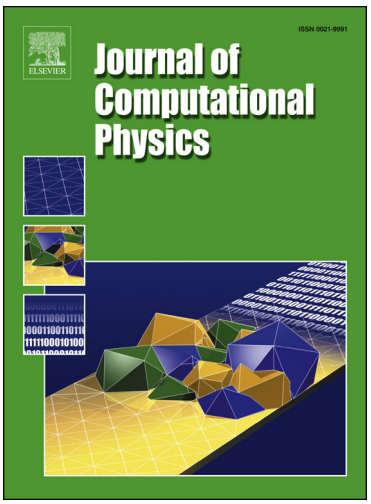
Accepted Manuscript



Kriging-sparse Polynomial Dimensional Decomposition surrogate model with adaptive refinement

Andrea F. Cortesi, Ghina Jannoun, Pietro M. Congedo

PII: S0021-9991(18)30718-6
DOI: https://doi.org/10.1016/j.jcp.2018.10.051
Reference: YJCPH 8353

To appear in: *Journal of Computational Physics*

Received date: 17 October 2017
Revised date: 3 May 2018
Accepted date: 31 October 2018

Please cite this article in press as: A.F. Cortesi et al., Kriging-sparse Polynomial Dimensional Decomposition surrogate model with adaptive refinement, *J. Comput. Phys.* (2018), https://doi.org/10.1016/j.jcp.2018.10.051

Highlights

- Polynomials selected by adaptive PDD are used as a sparse basis for Universal Kriging.
- Novel strategy to refine the Experimental design based on anisotropic mesh adaptation.
- Adding a fixed number of new training points on edges maximizing an error criterion.
- Convergence improvements on several algebraic test-cases with respect to the state-of-the-art.
- Application on two engineering problems in the aerospace field.

Kriging-sparse Polynomial Dimensional Decomposition surrogate model with adaptive refinement

Andrea F. Cortesi[a,*], Ghina Jannoun[b], Pietro M. Congedo[a]

[a] *INRIA Bordeaux Sud-Ouest, 200 Rue de la Vieille Tour, 33405 Talence, France*
[b] *CESI Engineering School, Nice Campus, 1240 Route des Dolines, 06560 Sophia Antipolis, France*

Abstract

Uncertainty Quantification and Sensitivity Analysis problems are made more difficult in the case of applications involving expensive computer simulations. This is because a limited amount of simulations is available to build a sufficiently accurate metamodel of the quantities of interest.

In this work, an algorithm for the construction of a low-cost and accurate metamodel is proposed, having in mind computationally expensive applications. It has two main features. First, Universal Kriging is coupled with sparse Polynomial Dimensional Decomposition (PDD) to build a metamodel with improved accuracy. The polynomials selected by the adaptive PDD representation are used as a sparse basis to build a Universal Kriging surrogate model. Secondly, a numerical method, derived from anisotropic mesh adaptation, is formulated in order to adaptively insert a fixed number of new training points to an existing Design of Experiments.

The convergence of the proposed algorithm is analyzed and assessed on

*Corresponding author. Tel.: +33(0)524574000
Email address: andrea.cortesi@inria.fr (Andrea F. Cortesi)

different test functions with an increasing size of the input space. Finally, the algorithm is used to propagate uncertainties in two high-dimensional real problems related to the atmospheric reentry.

Keywords: Surrogate Modeling, Universal Kriging, sparse Polynomial Dimensional Decomposition, Anisotropic Adaptive Meshing, Adaptive Refinement

1. Introduction

A wide range of applications in the field of applied mathematics and engineering rely on the numerical solution of complex mathematical models. Today, with the important advancements in numerical modeling and the increasing computer powers, highly accurate simulations of complex physical phenomena can be obtained, often at a price of a prohibitive computational cost. Moreover, the computational burden can dramatically rise when several evaluations with different configurations or different parameters are needed, for example in (stochastic) design optimization, Bayesian inference, Uncertainty Quantification (UQ) or Sensitivity Analysis (SA). A common practice in these fields is to perform a limited amount of exact evaluations of the solution and then use the obtained values to build a surrogate model, able to emulate the output of the complex model in other points than the observed ones.

The problem of building a cheap metamodel able to give an accurate approximation of the real function is not trivial. Several techniques have been explored in the literature. Generic and more application-oriented techniques were proposed. An ongoing effort is still performed to improve their accuracy

and efficiency, especially in the case of high-dimensional real-life problems, where many classical metamodeling methods still require a dramatic number of model evaluations (several hundreds or thousands). Some examples of surrogate models include polynomial response surfaces [1], Polynomial Chaos expansion[2, 3], Polynomial Dimensional Decomposition [4], Radial Basis Functions [5] and Kriging [6].

Kriging is massively used in global optimization and uncertainty quantification [7] [8, 9] [10] [11]. As pointed out in [12], often in practical applications, Kriging is mostly used in its basic configuration known as Ordinary Kriging, because of the lack of *a priori* knowledge about the main trends of the function of interest. In a recent work, Kersaudy et al. [12] proposed to combine Universal Kriging with a different metamodeling technique, known as LARS Polynomial Chaos, which is able to find a good basis function for the regression term, leading to a more accurate metamodel with the same size of the design of experiments. In another previous work, the blind-Kriging method [13] was developed. It shares some similarities with the work in [12], but employs a Bayesian selection algorithm for the selection of basis functions.

The first main contribution of this paper is to assess the potential interest in using sparse Polynomial Dimensional Decomposition [14] as a regression term of Universal Kriging, inspired by [12], where a LARS Polynomial Chaos has been used. Polynomial Dimensional Decomposition (PDD) [4] is a technique for building a hierarchical decomposition of a multivariate function. It directly relies on the well-known ANOVA functional decomposition [15, 16] and they both share a close structure. In this way, the PDD is able to give the priority to exploit low-order parameter interactions, following the prin-

ciple where low-order ANOVA component functions are dominant for most engineering cases. For this reason, PDD is preferred over Polynomial Chaos (PC) expansion in this work. In fact, as pointed out in [17], if a (stochastic) function is highly nonlinear, but contains rapidly diminishing interaction effects of multiple variables, the PDD approximation is expected to be more effective than the PC approximation, as the lower-variate interaction terms of the PDD approximation can be just as nonlinear by selecting appropriate values of maximum polynomial degree in the PDD. However, many more terms and expansion coefficients must be included in the PC approximation to capture such a high nonlinearity. In [14], the authors proposed an adaptive sparse implementation of the PDD that showed to produce good results for sensibility analysis, thus being able to efficiently capture the main trends of the function. The purpose of coupling between Kriging and sparse-PDD is then to start from the surrogate produced by the sparse-PDD, able to follow the main trends of the function, and exploit the Kriging to add a correction that improves the quality of the metamodel and leads to an interpolating metamodel, at least when not considering the nugget effect in the covariance of the Gaussian process. Of course the training cost of the coupled technique will be higher with respect to each single normal method, but for expensive real-world applications it would be lower than the cost of a single evaluation of the complex model.

Another aspect that has a significant impact on the final quality of the metamodel is the choice of the Design of Experiments (DoE) (or Experimental Design), i.e. the set of sampling points (or training points) on which the real model is evaluated, and that are used to train the surrogate. A method

for efficiently adding training points to the initial DoE can be useful to improve the accuracy without discarding the previous model evaluations and the information acquired on the function of interest and metamodeling error. Often in literature, for the construction of surrogate models, training points are chosen according to space-filling criteria. A straightforward way to generate an experimental design with this characteristic is Latin Hypercube Sampling (LHS) [18]. This kind of DoE is widely used with good results, for example within the well-known DACE algorithm [19]. Improvements have been performed in [20] by coupling the generation of a LHS design with optimality criteria for Kriging training points. Techniques have also been developed to increase the number of points of an already existing design while keeping is good space-filling properties, for example the nested Latin Hypercube [21] or quasi Monte Carlo sampling. Looking specifically at Kriging methods, different techniques have been investigated to create and enrich set of training points. For example, in [22], the Maximum Mean Squared Error (MMSE), Integrated Mean Squared Error (IMSE) or an entropy criteria are used to create and sequentially increment the design of experiment in such a way that is possible to optimize the metamodeling error, which is considered to be the variance of the Gaussian process. However, it is important to remind that the Kriging variance is a model-based estimate of the metamodeling error, and it is more an indicator of the good distribution of the training points in the domain. In some cases it is possible that it is not a good estimate of the true metamodeling error. Furthermore, it does not depend directly on the evaluations of the function of interest.

Sometimes, since the function of interest could present a behavior which is

just locally more difficult to represent by means of a surrogate model, it could be useful to rely on some adaptive technique able to add a certain amount of points to the initial set of training points in the most problematic regions. Finding an adequate way to use the information about the already evaluated training points to globally and adaptively improve the metamodel accuracy is not straightforward. Some examples of adaptive designs can be found in the literature. They are often focused on a special purpose or linked to specific techniques. For example, the well known EGO algorithm by Jones et al. [23] is used to find sequential designs for the optimization of a deterministic function by choosing at each step the point that minimizes the Expected Improvement (EI). Another example is in the work of Witteveen about Simplex Stochastic Collocation (SSC) [24–27], where an adaptive refinement strategy is proposed and developed to add simplex elements in the probability space.

In this context, the second major contribution of this paper is to introduce a mesh adaptation method which adds new points to the initial Experimental Design by using an estimator which exploits also the Kriging-Sparse PDD construction. The adaptation method adopted in this work exploits a mesh adaptation technique [28, 29], developed in the context of Computational-Fluid-Dynamics (CFD) applications, to derive an algorithm which adds new training points along the edges of the grid according to the optimization of an error criterion. Nodes are added to the grid, where the training function presents sharp gradients in order to better capture the variation of the function all over the domain. As a consequence, the proposed algorithm starts by measuring the error along the edges of the mesh as a function of the gradient variation. Then in order to quadratically minimize the total error it optimally

inserts new nodes along these edges under the constraint of a fixed number of nodes. Unlike derivative-free optimization methods [30, 31], the newly developed algorithm uses the information about the function smoothness in order to converge faster to the optimal solution. Moreover, an additional algorithm is proposed permitting to choose the number of points to add, which could have a practical interest if the computational power available is restricted. Note that the proposed mesh adaptation technique is eventually not constrained to a specific metamodeling technique or experimental design.

Several mesh adaptation techniques were proposed in the literature in the context of metamodeling. Among these methods one can find the Mesh Adaptive Direct Search (MADS) algorithm [30]. Although these methods provided good approximations for small dimensional problems, their convergence rate can be very slow for higher dimensions as they rely on the function values without fully exploiting the inherent smoothness of the objective function. Metamodeling node insertion techniques based on either global or local search algorithms can be found in the literature. Gutmann [32] proposed a node selection method that relies on the minimization of a "bumpiness" measure whereas Regis and Shoemaker [33, 34] proposed an optimization approach that starts from a feasible random set of points then takes small step sizes and search for the points that best represent the metamodel function values. Jakobsson et al. [35] proposed an adaptation method that controls the total uncertainty. Although these optimization techniques were based on a robust theoretical basis and successfully applied to expensive functions, they induce a relatively important computational cost. In [25], the authors derived a rigorous stopping criterion for h-refinement. The proposed method

relies on a robust mathematical foundation. However, two efficiency bottlenecks of these methods can be detected. First, they do not consider any control on the mesh complexity as a function of the available computational resources. Second, they do not take advantage of the directional aspect of the functional gradient and hence apply the same refinement in all the directions.

Compared to the above cited adaptation techniques, the proposed approach is fast and very simple to apply as it only requires the values of the function on the grid points and a fixed number of additional nodes imposed by the user. The latter criterion is an asset for efficiency and accuracy. Given the computational power at hand, the user can fix a certain maximal mesh complexity and the algorithm will adapt to provide the optimal accuracy with that number of nodes. This is an advantage with respect to other adaptation methods [17, 25] that rely on a desired accuracy which most of the time cannot be reached due to the lack of computational power or memory capacity. In that case the other approaches try to lower the imposed accuracy then restart the computations. Finally, the directional feature of the refinement as well as its multi-component nature are key assets of the developed method that make it outperform classical uniform, structured and isotropic mesh refinement techniques found in the literature [17, 25].

The numerical framework proposed in this paper for the construction of an accurate surrogate model, while having in mind applications where the function of interest is expensive to evaluate, is summarized in Figure 1 and Algorithm 1. Following this structure, the paper is organized as follows. In Section 2, a synthetic description of adaptive sparse PDD (2.1) and Universal Kriging (2.2) is provided. Note that in Appendix A some techniques

are recalled for the metamodel assessment, which have been used in this work. Then, Section 3 illustrates the approach proposed in this paper for the construction of a robust surrogate model. Section 3.1 describes the coupling between the PDD basis functions with the regression term in Universal Kriging. Afterwards, Section 3.2 describes the proposed adaptive sampling, which is further developed in Section 3.2.3 and 3.2.4. Some results of convergence on different test functions are presented in Section 4.1, while Section 4.2 is focused on the adaptive sampling technique. Finally, in Section 4.3, the algorithm is applied to two engineering problems in aerospace. Section 5 draws some conclusions and considerations on future developments.

Algorithm 1 Global metamodeling procedure

```
if extrapolation (3.2.4) then
    Create an initial set of training points (DoE) in the input space
else
    Create an initial DoE in the input space + corners
end if
while unacceptable metamodeling error do
    Evaluate the function of interest in the training points
    Build the sparse-PDD metamodel to get the basis functions (2.1)
    Build the coupled PDD-UK metamodel (3.1)
    Estimate the metamodeling error
    if adaptation then
        Choose whether the number of added points should be fixed (3.2.3)
        or not (3.2.1)
        Tune the weight between Kriging and gradient-based error (3.2.2)
        Adaptively refine the DoE (with extrapolation if necessary)
    end if
end while
```

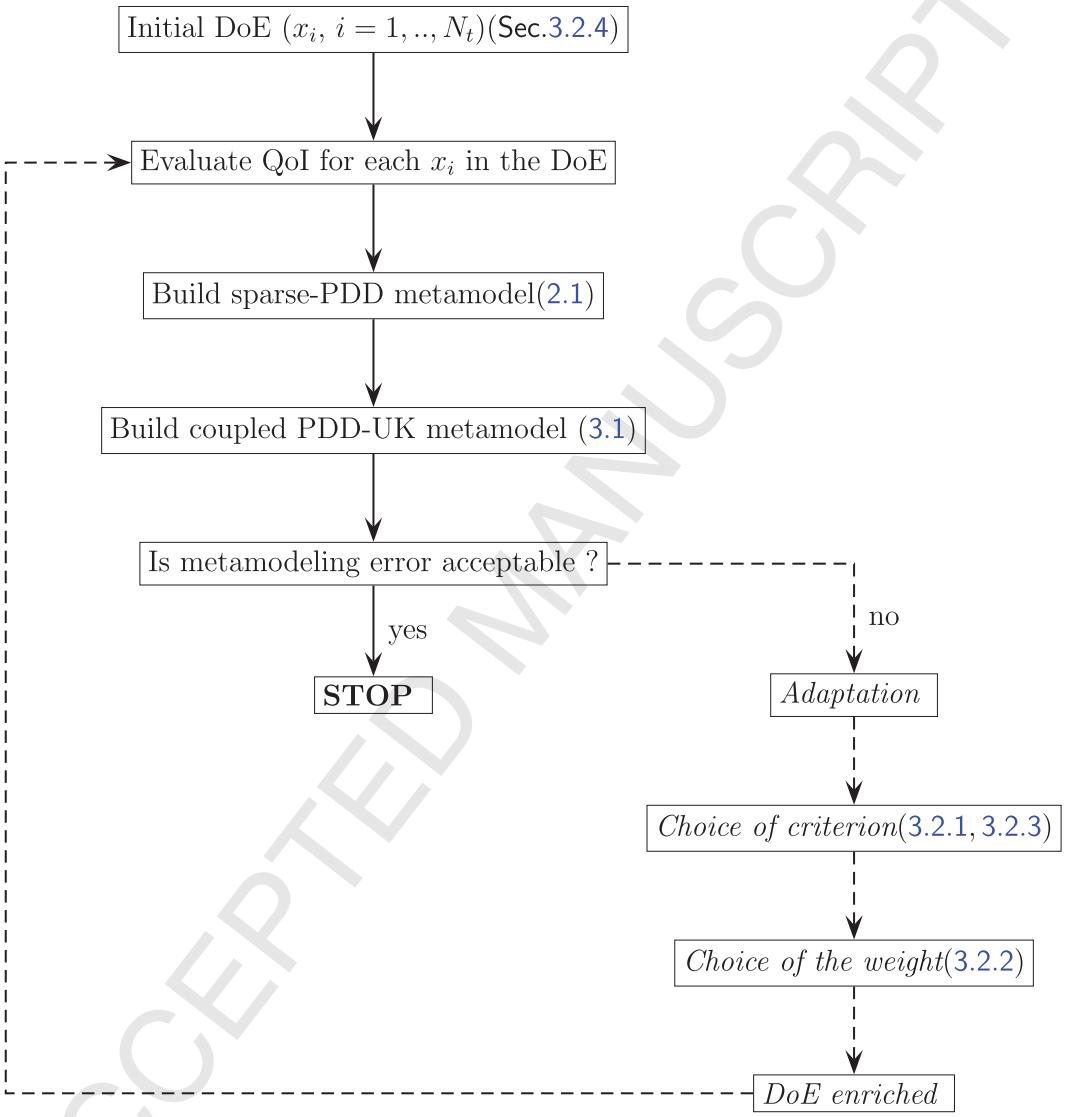


Figure 1: Global metamodeling framework

2. Starting point in metamodeling

Let us suppose to have a N dimensional input random variable $\boldsymbol{X} = \{X_1, \ldots, X_N\} \in \mathbb{R}^N$, with a joint probability density function (PDF) $p_{\boldsymbol{X}}(\boldsymbol{x})$. The assumption of independence of the components of this random vector implies that its PDF can be written as

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{N} p_{X_i}(x_i) \tag{1}$$

where $p_{X_i}(x_i)$ is the marginal PDF of X_i.

Let us suppose that the response of a given system can be represented by a N-dimensional function of interest $y = f(\boldsymbol{x})$. In this work, the quantities of interest are considered to be scalar, *i.e.* $y \in \mathbb{R}$. Different engineering problems, such as optimization or uncertainty quantification, may require several evaluations of the function of interest at different values of $\boldsymbol{x}$. However, this may be very expensive to be evaluated, for example in complex computational models based on partial differential equations. Therefore, in this cases, one can resort to using a cheap surrogate model $\hat{f}(\boldsymbol{x})$ to predict the value of the quantity of interest. The quantity of interest can be treated as a noiseless quantity, since it is the output of a deterministic computational model considered to be exact.

2.1. Regression-based adaptive sparse-PDD

In this section, the sparse-Polynomial Dimensional Decomposition (sPDD) technique implementation, used in this work, is recalled. This adaptive strategy to build a PDD metamodel with a sparse approach has been proposed

by Tang et al. in a recent work [14]. In the original paper, the technique addresses primarily problems of sensitivity analysis and uncertainty quantification, but it can be employed also as a surrogate model representation of the function of interest for other applications.

2.1.1. Classical PDD representation

Let us briefly recall the ANOVA representation of a multivariate function. More details can be found, for example, in [15, 16]. In general, the multivariate function of interest can be represented by the following expansion:

$$y = f(\boldsymbol{x}) = f_0 + \sum_{s=1}^{N} \sum_{i_1<\cdots<i_s}^{N} f_{i_1 \ldots i_s}(x_{i_1}, \ldots, x_{i_s}) \tag{2}$$

which can be rewritten in compact form exploiting a multi index notation:

$$y = f_{s_0} + \sum_{j=1}^{\mathcal{M}} f_{s_j}(\boldsymbol{x}_{s_j}) \qquad \text{with} \quad \mathcal{M} = 2^N - 1 \tag{3}$$

This representation is called ANOVA (Analysis of Variance) decomposition if, for any $j \in 1, \ldots, \mathcal{M}$, the following orthogonality condition is respected

$$\int_{\mathbb{R}} f_{s_j}(\boldsymbol{x}_{s_j}) p_{X_i}(x_i)\, dx_i = 0 \qquad for \quad x_i \in \{\boldsymbol{x}_{s_j}\} \tag{4}$$

Some useful properties of the ANOVA decomposition are shown in [14].

Until this point, the component functions f_{s_j} of the ANOVA decomposition of the function of interest are not determined yet. In literature, two techniques are mainly used for this purpose: Polynomial Chaos expansion (PC) and Polynomial Dimensional Decomposition (PDD). As shown in

[14, 17], the PDD could be preferred for its closer structure with respect to ANOVA.

Let us consider then an orthogonal set of polynomials in the Hilbert space $\mathcal{L}_2$, denoted by $\{\psi_j(x_i); j = 0, 1, \dots\}$, such that

$$\int_{\mathbb{R}} \psi_j(x_i)\psi_k(x_i)p_{X_i}(x_i)\,dx_i = \gamma_{j,X_i}\delta_{ij} \tag{5}$$

where j and k are the polynomial orders for the variable x_i and γ_{j,X_i} the normalization constant determined by

$$\gamma_{j,X_i} = \int_{\mathbb{R}} \psi_j^2(x_i)p_{X_i}(x_i)\,dx_i. \tag{6}$$

As well-known in literature (see [4] for example), an optimal choice is to have a basis corresponding to families of polynomials which are orthogonal to some probability distributions.

Let us consider now a T-dimensional term of the ANOVA decomposition, with $1 \leq T \leq N$

$$f_{i_1,i_2,\dots,i_T}(x_{i_1}, x_{i_2}, \dots, x_{i_T}) \tag{7}$$

The component function can be expanded as done in [4]:

$$f_{i_1,i_2,\dots,i_T}(x_{i_1}, x_{i_2}, \dots, x_{i_T}) = \sum_{j_T=1}^{\infty} \cdots \sum_{j_1=1}^{\infty} C_{i_1,i_2,\dots,i_T}^{j_1,j_2,\dots,j_T} \prod_{k=1}^{T} \psi_{j_k}(x_{i_k}) \tag{8}$$

In practice the expansion must be truncated, leaving m terms for each dimension. Thus, the polynomial dimensional decomposition of the function

of interest can be written in the following final expression

$$f(\boldsymbol{x}) \simeq \hat{f}_m(\boldsymbol{x}) = f_0 + \sum_{i=1}^{N}\sum_{j=1}^{m} C_i^j \psi_j(x_i) + \sum_{i_1<i_2}^{N}\sum_{j_2=1}^{m}\sum_{j_1=1}^{m} C_{i_1,i_2}^{j_1,j_2} \psi_{j_1}(x_{i_1})\psi_{j_2}(x_{i_2}) + \\ + \sum_{i_1<i_2<i_3}^{N}\sum_{j_3=1}^{m}\sum_{j_2=1}^{m}\sum_{j_1=1}^{m} C_{i_1,i_2,i_3}^{j_1,j_2,j_3} \psi_{j_1}(x_{i_1})\psi_{j_2}(x_{i_2})\psi_{j_3}(x_{i_3}) + \\ + \cdots + \sum_{i_1<\cdots<i_N}^{N}\sum_{j_N=1}^{m}\cdots\sum_{j_1=1}^{m} C_{i_1,i_2,\ldots,i_N}^{j_1,j_2,\ldots,j_N} \prod_{k=1}^{N} \psi_{j_k}(x_{i_k}) \qquad (9)$$

Hence the total size P of the m-th order PDD expansion of an N-dimensional function is $P = (1+m)^N$. As in [14], a regression approach is preferred to a projection approach for the computation of the expansion coefficients $C_{\cdots}^{\cdots}$. This first approach is supposed to be more flexible for problems with a moderate number of variables, but the corresponding surrogate model does not interpolate exactly the training points.

2.1.2. Dimension reduction for the model representation

For practical problems, in particular the ones with a moderate to large number of stochastic variables, the size of the PDD expansion becomes very large. For this reason, in [14] an adaptive dimension reduction of the representation has been proposed. This adaptive technique belongs to the family of *stepwise regression*, and has been inspired by several previous papers [36–38].

Since the lower order interaction terms often have the greater impact on the output [16], the full ANOVA expansion can be truncated at a maximum

dimension of component functions $\nu < N$, called the truncation dimension

$$f(\boldsymbol{x}) = f_0 + \sum_{T=1}^{\nu} \sum_{i_1<\cdots<i_T}^{N} f_{i_1,i_2,\ldots,i_T}(x_{i_1}, x_{i_2}, \ldots, x_{i_T}) \tag{10}$$

However, especially for problems with a high stochastic space's dimension N, ANOVA decomposition can still be very expensive. This problem can be addressed by using the adaptive ANOVA decomposition

$$f(\boldsymbol{x}) = f_0 + \sum_{T=1}^{\nu} \sum_{i_1<\cdots<i_T}^{D_T} f_{i_1,i_2,\ldots,i_T}(x_{i_1}, x_{i_2}, \ldots, x_{i_T}) \tag{11}$$

where $D_T < N$ is the active dimension of the component function of T-th order. In this work, we will consider $D_1 = N$ and the active dimension for higher order terms will be determined with the criterion proposed afterwards. This formulation is closely related to the one introduced in [38]. Other formulations for the adaptive ANOVA can be found, for example, in [39–41].

Even with an adapted ANOVA expansion, the computational cost required to compute the classical PDD expansion for each component function is still very high. However, in many real-world problems the contribution of some polynomial terms is negligible with respect to the accuracy of the metamodel or their impact on the global variance. This fact can be exploited to build a sparse PDD representation without compromising the accuracy of the metamodel.

The global adaptive sparse algorithm can be described combining adaptive ANOVA and sparse PDD [14]:

1. Construct a full set of PDD representation (given m) for all the first order ANOVA component functions

$$f(\boldsymbol{x}) \simeq f_0 + \sum_{i=1}^{N} f_i(x_i) = f_0 + \sum_{i=1}^{N}\sum_{j=1}^{m} C_i^j \psi_j(x_i). \tag{12}$$

   Compute then the total first order variance (see [14])

$$\mathrm{Var}(f(\boldsymbol{x})) = \sum_{i=1}^{N}\sum_{j=1}^{m} (C_i^j)^2 \gamma_i^j \tag{13}$$

   and first order Sobol' sensitivity indices [15, 16]

$$\mathcal{S}_i = \frac{\sum_{j=1}^{m} (C_i^j)^2 \gamma_i^j}{\mathrm{Var}(f(\boldsymbol{x}))} \tag{14}$$

   Assuming the sensitivity indices to be monotonically decreasing ordered with respect to i, it is possible to choose the active dimension D_2 for second order ANOVA functions such as $\sum_{i=1}^{D_2} \mathcal{S}_\rangle \geq p$.
2. Reduce the size of the first order PDD expansion expressed in equation 12, eliminating the less important terms in variance contribution, exploiting one of the two selection criteria described later. The obtained first-order reduced basis is denoted by $\{\psi_{\alpha^1}\}$.
3. Enrich the model by adding significant higher order terms at the concise first-order basis, obtaining the final basis $\{\psi_{\alpha^F}\}$.

In [14], two different algorithms were described to choose the most relevant terms to be added to the sparse basis. In the first (see algorithm 2), that was introduced in [14], the terms are chosen according to their contribu-

Algorithm 2 Variance-based adaptive PDD by stepwise regression

Initialization of PDD basis $\{\psi^w\} = \{\psi_{\alpha^1}\}$
for $\psi_{\alpha_i} \in \{\psi_{\alpha^{2+}}\}$ **do**
  add ψ_{α_i} into $\{\psi^w\}$, namely $\{\psi^w\} = \{\psi^w, \psi_{\alpha_i}\}$
  solve the regression system to determine the PDD expansion coefficients
  compute the total variance $\mathrm{Var}(f^w(\boldsymbol{X}))$
  for $\psi_{\alpha_j} \in \{\psi^w\}$ **do**
    if $(C_{\alpha_j})^2\gamma_{\alpha_j}/\mathrm{Var}(f^w(\boldsymbol{X})) < \epsilon$ **then**
      eliminate this polynomial: $\{\psi^w\} = \{\psi^w\}\backslash\psi_{\alpha_j}$
    end if
  end for
end for
solve the final regression system based on the constructed basis $\{\psi^F\}$

tion to the total variance associated to the output function of interest. This algorithm proved to be very effective for the computation of the sensitivity indices. The second algorithm, taken from previous works [36, 37] on sparse Polynomial Chaos, is based on the metamodeling error (see algorithm 3) and instead selects the most relevant terms according to their contribution to the global metamodeling error computed with a leave-one-out cross-validation. Another error-based adaptive-sparse PDD implementation can be found, for example, in [17].

All the assumptions done to produce an adaptive ANOVA representation and a sparse PDD regression can of course be inaccurate in some applications, by yielding a surrogate model with a poor quality. However, in general, if the user-defined parameters are properly tuned, the adaptive-sparse surrogate should be able to produce an accurate representation of the function of interest in a more efficient way with respect to a full representation.

Algorithm 3 Error-based adaptive PDD by stepwise regression

Initialization of PDD basis $\{\psi^w\} = \{\psi_{\alpha^1}\}$
for $\psi_{\alpha_i} \in \{\psi_{\alpha^{2+}}\}$ **do**
  add ψ_{α_i} into $\{\psi^w\}$, namely $\{\psi^w\} = \{\psi^w, \psi_{\alpha_i}\}$
  solve the regression system to determine the PDD expansion coefficients
  evaluate the metamodel accuracy Q_i^2
  if $Q_i^2 \geq Q_{tgt}^2$ **then**
    exit
  end if
  for $\psi_{\alpha_j} \in \{\psi^w\}$ **do**
    solve the regression system with the polynomial basis $\{\psi^w\} \backslash \psi_{\alpha_j}$
    evaluate the metamodel accuracy $Q_{i\backslash\alpha_j}^2$
    if $Q_i^2 - Q_{i\backslash\alpha_j}^2 < \epsilon$ **then**
      eliminate this polynomial: $\{\psi^w\} = \{\psi^w\} \backslash \psi_{\alpha_j}$
    end if
  end for
end for
solve the final regression system based on the constructed basis $\{\psi^F\}$

2.2. Universal Kriging surrogate

Kriging interpolation [6, 19, 22, 42] is a well-known technique for building a surrogate model. Its main idea is to consider the function of interest $f(\boldsymbol{x})$ as a realization of a stationary Gaussian stochastic process $F(\boldsymbol{x})$. In Universal Kriging (UK), the stochastic process can be written in the form of the sum of a deterministic regression model and a stochastic departure term

$$F(\boldsymbol{x}) = \sum_{j=1}^{n} \beta_j y_j(\boldsymbol{x}) + Z(\boldsymbol{x}) \tag{15}$$

where $y_j(\boldsymbol{x})$ are linearly independent known regression functions, β_j are unknown weights, and $Z(\boldsymbol{x})$ is a Gaussian process with zero mean and covari-

ance $k(\boldsymbol{u}, \boldsymbol{v})$. The departure term is assumed to be correlated as a function of the distance between different points, using a prescribed correlation model such as exponential, Gaussian or Matèrn correlation functions [43]. The parameters σ and $\boldsymbol{l} = \{l_j\}_{j=1}^{N}$, representing respectively the amplitude of the correlation and the correlation lengths, are called *hyperparameters* and enter as known parameters in the construction of the Kriging response surface. For this reason they need to be estimated in a previous step, for example by using the maximum likelihood estimation.

Then the final aim is to build an interpolation of the function of interest in the form

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{N_s} f(\boldsymbol{x}_i)\lambda_i(\boldsymbol{x}) \tag{16}$$

where $\boldsymbol{f}_{obs} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_{N_s}))^T$ are the observations of the function at the N_s training points (the design of experiment) and $\lambda_i(\boldsymbol{x})$ are unknown weights. The best linear unbiased predictor can be obtained by minimizing the mean square error between the model and the predictor $MSE = E[(F(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}))^2]$ under the constraint of unbiasedness $E[F(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})] = 0$. In this way it is possible to obtain the predictive mean of the stochastic process, that can be used as a metamodel for the original function

$$\begin{aligned} f(\boldsymbol{x}) \simeq \hat{f}(\boldsymbol{x}) = \mu_k(\boldsymbol{x}) = \boldsymbol{y}^T(\boldsymbol{x})\boldsymbol{\beta} + \boldsymbol{c}(\boldsymbol{x})^T C^{-1}(\boldsymbol{f}_{obs} - Y\boldsymbol{\beta}) \\ \text{with} \quad \boldsymbol{\beta} = (Y^T C^{-1} Y)^{-1} Y^T C^{-1} \boldsymbol{f}_{obs} \end{aligned} \tag{17}$$

where $\boldsymbol{y}(\boldsymbol{x}) = (y_1(\boldsymbol{x}), \ldots, y_n(\boldsymbol{x}))^T$ is the vector of basis functions, Y is a matrix whose elements are the evaluation of the j-th basis function at the i-th training point $Y_{ij} = y_j(\boldsymbol{x}_i)$, $\boldsymbol{c}(\boldsymbol{x})$ is the vector of correlations between the

point $\boldsymbol{x}$ and each training point and C is the matrix of correlations among training points.

It is possible to compute also the predictive variance, which can be used as a local model-based error estimate

$$\sigma^2(\boldsymbol{x}) = k(\boldsymbol{x},\boldsymbol{x}) + \boldsymbol{a}(\boldsymbol{x})^T (Y^T C^{-1} Y)^{-1} \boldsymbol{a}(\boldsymbol{x}) - \boldsymbol{c}(\boldsymbol{x})^T C^{-1} \boldsymbol{c}(\boldsymbol{x}) \qquad (18)$$
$$\text{with} \quad \boldsymbol{a}(\boldsymbol{x}) = Y^T C^{-1} \boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{y}(\boldsymbol{x})$$

In real-world applications, often the Universal Kriging is not used, because it can be difficult to determine relevant basis functions for the regression term in Equation 15 without the proper *a priori* knowledge about the evolution of the quantity of interest. Hence, one is limited to use the simpler technique called *Ordinary Kriging*, in which the regression functions are chosen as $y_1(\boldsymbol{x}) = 1$ and $y_j(\boldsymbol{x}) = 0$ for $j \neq 1$, which means that only a constant regression term is kept and then only β_1 needs to be determined. This simplifies the method but can limit the accuracy of the metamodeling technique, thus requiring a higher number of training points in order to obtain a representation of the output function with a certain level of accuracy.

3. PDD-UK surrogate model and adaptive sampling

This section illustrates the two algorithms proposed in this paper, which are mentioned in Algorithm 1, for the construction of an accurate surrogate model, while having in mind applications where the function of interest is relatively expensive to evaluate, involving for example the solution of a system of partial differential equations.

First, in Section 3.1, we provide the description of an improved coupled PDD-Universal Kriging surrogate. The goal is to reduce the number of training points required to achieve a desired accuracy. Secondly, a refinement procedure is introduced for the experimental design (see Sections 3.2 and 3.2.3). It consists in treating the training points as nodes of a simplex grid and applying an adaptation method derived from anisotropic adaptive meshing domain. If the estimated metamodeling error exceeds a desired threshold, the refinement procedure is applied, and the Design of Experiments is further enriched with new samples.

3.1. Construction of the coupled PDD-UK surrogate

As pointed out by Kersaudy *et al.* in [12], often the lack of *a priori* knowledge on the function of interest forces the use the Kriging technique in the basic configuration known as Ordinary Kriging (see Section 2.2). They proposed to use a sparse polynomial chaos (PC) expansion computed with LARS algorithm to obtain a set of regression functions to build an Universal Kriging surrogate model.

In this paper, the basis is chosen by applying the adaptive sparse PDD algorithm (Sec. 2.1). This is a very sparse representation of the function of interest that is able to achieve a good metamodeling accuracy and to discard the stochastic variables whose influence on the output value is negligible. Then, the basis is used as a regression function for the Universal Kriging surrogate model. The use of the most influential polynomials as basis for the Universal Kriging should improve the quality of the final surrogate by adding the most relevant information about the trends of the quantity of interest to the regression term.

The algorithm used here to couple Universal Kriging with the sparse-PDD basis functions is the following:

1. Build a design of experiments (set of training points)
2. Perform an adaptive sparse PDD and obtain a set of relevant basis function $\{\psi_{\alpha^F}\}$, with the following PDD representation of the function of interest:

$$\hat{f}(\boldsymbol{x}) = f_0 + \sum_{\alpha \in \alpha^F} C_\alpha \psi_\alpha \qquad \text{with} \quad C_\alpha \psi_\alpha = C_{i_\alpha}^{j_\alpha} \psi_{j_\alpha}(x_{i_\alpha}) \tag{19}$$

   where i_α and j_α are the respective multi-indices. Moreover, $\psi_0 = 1$ must be kept in the set of basis function for the Universal kriging.
3. Train the Kriging surrogate, within a Universal Kriging approach, using the basis functions of the sparse PDD expansion as regression functions

$$F(\boldsymbol{x}) = \sum_{j \in \alpha^F} \beta_j \psi_j + Z(\boldsymbol{x}) \tag{20}$$

It must be noticed that the original coupling algorithm of Kersaudy and coworkers is more complex. In fact, it builds a UK metamodel at each step of the cycle used to enrich the sparse PC basis and computes cross validation error [44] for each metamodel, then choses the best metamodel according to the error. Even if this approach could lead to further improvements of the quality of the metamodel, we have chosen here a lighter formulation based on the selection of the controlling parameters suggested in [14], followed by a systematic check about the influence of the controlling parameters. However, this difference in the implementation does not change the main idea of the

coupling process, and the same strategy proposed in [12] can be applied to the proposed framework.

Note also that in cases where just a subset of the input variables contributes for the most to the variation of the output, it could be possible in the proposed framework to reduce the size of the input to facilitate the training of the surrogate model, thus improving its quality. The simple strategy that can be exploited in the framework proposed in this work consists in building the final UK considering as input variables just the ones whose total sensitivity index, computed with sparse-PDD, is non-null. In this way all the inputs which show an irrelevant contribution to the output are neglected, simplifying the training problem and the fitting of the hyperparameters. Other techniques for the input size reduction and their coupling with Kriging surrogate models have been developed in literature. Two examples are the Active Subspaces method [45] and anchored-ANOVA [46], but the in-deep analysis and comparison are not object of study in this paper. The dimension reduction strategy test will be just proposed for the TACOT ablation engineering case (see 4.3.2).

Finally, we would like to remind that the targeted quantities of interest in this work are the outcome (normally a scalar value) of expensive computer codes, which is assumed to be exact. Therefore, each QoI is assumed to be a noiseless quantity throughout the whole manuscript. However, the extension of the proposed framework to the treatment of noisy outcomes should be straightforward. In fact, the PDD-UK is still a Universal Kriging surrogate model, and therefore it is possible to account for noisy QoIs by following the approach commonly used in Kriging, *i.e.* by exploiting the so-called

nugget effect [47]. Also, being a Kriging surrogate, it can be used in all the applications involving the use of the predictive variance, like IMSE and MMSE [22, 48] for the adaptation of the design of experiment or the Expected Improvement [23] for the Bayesian optimization.

3.2. Adaptive sampling through anisotropic mesh adaptation

One critical aspect in the accuracy of metamodels is the construction of a good set of training points. Often in literature, points are chosen according to space-filling criteria, such as Latin Hypercubes designs. However, in some applications, the total number of available evaluations of the accurate model can be limited by its elevated computational cost, and a first surrogate model built on this small design of experiments could show a lower accuracy than desired. In this section, we present a method for efficiently adding training points to the initial DoE, without discarding the previous model evaluations and exploiting the information acquired on the function of interest and metamodeling error.

3.2.1. Basic Algorithm

The adaptation method proposed in this work is based on building a mesh of simplex elements in the stochastic space of the input parameters, considering the training points as nodes of the elements, and on exploiting an extended mesh adaptation technique, derived from CFD applications.

In particular, new training points are added along the edges of the grid according to an error optimization criterion. The main drawback of this technique is the fact that it relies on the notion of edges of a mesh, and, as known, the construction of an n-dimensional Delaunay triangulation for

higher dimensional spaces can become very costly.

A first simple implementation of the node insertion algorithm is directly derived from the work of Coupez et al. [28, 29] and Jannoun [49] on anisotropic adaptive meshing. In the original algorithm, the total number of mesh nodes is fixed by the user and does not change during the adaptive process. The algorithm controls the edge error while respecting a fixed number of nodes in the mesh. From the latter, a threshold global error is computed and used to compute stretching and shrinking factors to adapt the mesh and move the existing nodes. In this work, instead, the interest is basically in adding new points to the existing mesh. Unlike the method in [28, 29, 49], old mesh nodes will not be allowed to moved all over the domain. Therefore, as will be described in what follows, the algorithm will be adapted to a mesh node insertion approach.

3.2.2. Error criterion

The error criterion, called e_k, is quadratically defined on each edge $k = 1, \ldots, n_e$ as the projected gradient on that edge. Hence, the edge based error estimation is expressed as the following quantity:

$$e_k = \sqrt{\sum_{j=1}^{n_{dim}} \left((g_j(\boldsymbol{x}_k^{(2)}) - g_j(\boldsymbol{x}_k^{(1)}))(x_{k_j}^{(2)} - x_{k_j}^{(1)}) \right)^2} \tag{21}$$

where $\boldsymbol{x}_k^{(1)}$ and $\boldsymbol{x}_k^{(2)}$ are the coordinates of the two nodes of a given edge, and $g(.)$ is the gradient of the function of interest. In practice, since the actual gradient usually is not known in this case, except for adjoint-based deterministic solvers, one could exploit the surrogate to compute numerical

gradients at the training points. Then, a stretching factor s_k can be defined for each edge, starting from the computed error value e_k and a target value e_{target}:

$$s_k = \left(\frac{e_{\text{target}}}{e_k} \right)^{\frac{1}{2}}. \tag{22}$$

If the stretching factor is smaller than one, it means that the error is higher than the target value, hence it is necessary to add a number n_k of nodes evenly spaced along the edge to reduce the local error. This number is computed as:

$$n_k = \left\lfloor \frac{1}{s_k} \right\rfloor = \lfloor N_k \rfloor \tag{23}$$

where $\lfloor N_k \rfloor$ denotes the *floor* of the quantity N_k, meaning the closer integer smaller or equal than N_k. The resulting algorithm is described in Algorithm 4. This procedure can then be repeated several times in an iterative cycle to increase sequentially the number of added training points.

Algorithm 4 Basic algorithm

```
Build the triangulation
Compute e_k for each edge k = 1 ... n_e
compute s_k = (e_target / e_k)^{1/2} and N_k = 1 / s_k
for each edge k do
    if ⌊N_k⌋ > 0 then
        add ⌊N_k⌋ evenly spaced new points along the edge
        add the new points to the DoE
    end if
end for
```

In order to enrich the mesh construction approach, it is useful to add some information about the accuracy of the metamodel to the error estimator, since gradients are computed numerically starting from the metamodel itself.

Therefore, it is interesting to refine also in regions where the gradient is low but the metamodeling error is high. Hence a weighted combination of Kriging variance and gradient error indicators would give a higher accuracy. A simple local-based error estimator for Kriging surrogate models is the Gaussian process variance. However, this indicator is zero at the training points, since the metamodel is an interpolation (in absence of nugget effect) and at these points we know the exact function value. A possible way to take it into account in the computation of the error e_k is to consider its value $\sigma_k = \sigma(\boldsymbol{x}_k^c)$ computed at the midpoint of the edge $\boldsymbol{x}_k^c = (\boldsymbol{x}_k^{(1)} + \boldsymbol{x}_k^{(2)})/2$. Then it could be re-scaled and summed to the existing gradient based criteria e_k, defined in Eq. 21, resulting in the following weighted sum:

$$e_k^{\text{weighted}} = \alpha\, e_k + (1-\alpha)\frac{\max_k (e_k)}{\max_k (\sigma_k)}\,\sigma_k \tag{24}$$

where α is an adjustable weight that controls the relative contribution of the two criteria to the global error on the edge.

3.2.3. Refinement by adding a fixed number of points

Having the possibility to add a fixed number of training points at each iteration of the adaptive process can be an advantage, because, in this way, it is easier for the user to parallelize, according to the available resources, the task of computing the actual value of the function of interest, which can involve expensive simulations. Two different strategies for the implementation of this feature are available. The first is a more rigorous mathematical formulation of the problem, but it translates in a more difficult and expensive algorithm, while the second tackles the problem directly from a numerical

point of view resulting in a faster algorithm. The two approaches are detailed in what follows and are named as the *Brute Approach* and the *Fast approach* (edge-based length distribution method), respectively.

The *Brute Approach* is a first rigorous attempt to node insertion. It consists in formulating the adaptation problem as an optimization one, that looks for the best combination of positions of a fixed number N_a of new nodes on the n_e edges in order to minimize an imposed error criterion. We seek to solve the following optimization problem:

$$N_k = \arg\min(\varphi(N_k)) \qquad \text{with} \qquad \varphi(N_k) = \sum_{k=1}^{n_e} \frac{1}{2} \frac{e_k}{(N_k+1)^2} \tag{25}$$

$$\text{with the constraint:} \quad N_a = \sum_{k=1}^{n_e} N_k$$

where e_k is the edge error and N_k is the number of nodes added to that edge. This optimization problem can be quite tricky to solve, due to the discrete nature of the design variables ($N_k \in \mathbb{N}$). However, it can be noticed that, when $N_a < n_e$, it is possible to consider in the optimization just the N_a edges associated to a higher error value, so the problem becomes

$$N_k = \arg\min(\varphi(N_k)) \qquad \text{with} \qquad \varphi(N_k) = \sum_{k=1}^{N_a} \frac{1}{2} \frac{e_k}{(N_k+1)^2} \tag{26}$$

$$\text{with the constraint:} \quad N_a = \sum_{k=1}^{N_a} N_k$$

At this point, a possible approach to solve the problem through the use of brute force consists in seeking among all possible permutations of N_a nodes on N_a edges the one that minimizes $\varphi(N_k)$. As it can be easily detected, this

approach is convenient just for small enough values of N_a (i.e. $N_a \leq 10$), since the number of total cases to be explored n_t increases quickly:

$$n_t = \frac{(N_a + N_a - 1)!}{(N_a - 1)! N_a!} \tag{27}$$

The method showed to be powerful and very accurate for low dimensions problems. However, it might be difficult to implement and expensive to solve for increasing size of the dimensional space.

The second approach, *i.e.* the *Fast approach*, consists in rewriting the optimization problem and modifying the iterative cycle of the original code in 3.2.1 so that it adds, at each iteration, a fixed number of points N_a imposed by the user. The implementation of the algorithm is described in Algorithm 5. While this more empirical insertion technique does not necessarily converge (from a strictly mathematical point of view), to the optimal rigorous solution, it highly decreases the computational cost, especially for higher values of N_a.

The nodes insertion method is adapted from the work in [28, 29, 49] whereby an anisotropic mesh adaptation technique was introduced. The original method consists in computing a stretching factor associated with each edge in the mesh in the view of minimizing an error criterion over the mesh while respecting a certain fixed number of nodes. It starts by evaluating the edge based error estimates in terms of the gradient of the function under consideration. In this work, since the focus is only on inserting new training nodes without moving the already existing ones, an adaptation of the original algorithm is proposed.

The optimization problem can be stated also with respect to the stretch-

ing factors s_k associated to the edges of the mesh:

$$\begin{aligned} &\text{minimize} \qquad \varphi(s_k) = \sum_{k=1}^{n_e} \frac{1}{2} s_k^2 e_k \\ &\text{subject to} \quad N_a = \sum_{k=1}^{n_e} \lfloor s_k^{-1} \rfloor \end{aligned} \tag{28}$$

since the number of points added on each edge can be related to the stretching factors (and hence to the error estimates) through the relation $N_k = \lfloor s_k^{-1} \rfloor$. An illustration of the node insertion approach along an edge $\mathbf{C}^{ij}$ is presented in figure 2. It can be noticed, however, that also in this case, although the design variable is continuous, in the constraint there is a function that transforms it into an integer, and so the same difficulties present in the previous approach arise. In practice, this constraint is relaxed. Therefore, the strong formulation is replaced by a weak formulation defined by:

$$\begin{aligned} &\text{minimize} \qquad \varphi(s_k) = \sum_{k=1}^{n_e} \frac{1}{2} s_k^2 e_k \\ &\text{subject to} \quad A = \sum_{k=1}^{n_e} s_k^{-1} \end{aligned} \tag{29}$$

where $A = N_t + N_a$, and N_t is the number of current training points. Then it is possible to find a solution by using a Lagrangian formulation

$$\mathcal{L}(s_k, \lambda) = \frac{1}{2} \sum_{k=1}^{n_e} s_k^2 e_k + \lambda \Big(\sum_{k=1}^{n_e} s_k^{-1} - A \Big), \tag{30}$$

which leads to

$$s_k = \Big(\frac{\lambda}{e_k} \Big)^{\frac{1}{3}}, \tag{31}$$

with the multiplier

$$\lambda = \left[\frac{\left(\sum_{k=1}^{n_e} e_k^{\frac{1}{3}} \right)}{A} \right]^3 . \tag{32}$$

Clearly, the generated solution does not necessarily respect the exact number of nodes as a truncation to the integral part of s_k^{-1} is applied. This process adds new points where it is most needed i.e. where the error is most important. Hence the choice of considering the floor of s_k^{-1}. As a result, the solution of problem (29) results in an adaptation algorithm that adds a number of points close but not necessarily equal to N_a since the strong formulation is replaced by a weaker one. Therefore, in order to meet the fixed number of nodes, a correction part has been implemented in Algorithm 5. It consists in repeating the cycle of node insertion where the estimated error is most important as long as the target number of nodes has not been reached.

Due to the quadratic nature of the optimization problem and the weighting of the gradient variation by the lengths of the edges in the computation of the error estimates, the method equally distributes the added nodes on the edges presenting the highest gradient variation. In other words, the method will not concentrate all the added points on one edge while keeping the other edges unrefined. It will instead start by distributing the nodes in the decreasing order of the computed error. Then, if the number of nodes fixed by the user is not reached, the method will iterate again on the edges adding more points to the ones that present the highest errors. This equi-distribution process helps in avoiding clustering problems.

Several adaptive mesh refinement techniques for QoI error evaluation based on a posteriori and goal oriented error estimation can be found in

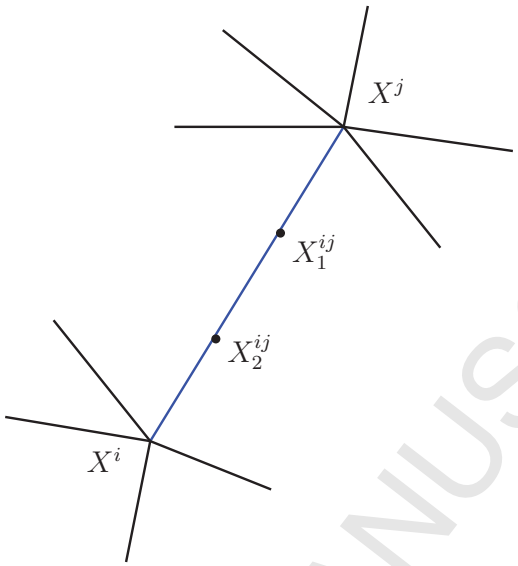


Figure 2: Adding $\lfloor N(k) \rfloor = 2$ evenly spaced new points along the edge $\mathbf{X}^{ij}$, where k denotes the edge number.

the literature [50–56]. These methods rely on an estimation associated with a particular QoI. Nevertheless, in many physical problems, there are several quantities of interest that need to be accounted for in a single adaptive algorithm. In order to adapt the mesh to the different parameters using these methods one has to compute an adjoint solution per quantity of interest which is computation-wise very expensive. In the view of reducing the computational cost induced by these methods, extensions of this approach to take into account several QoI have been proposed in [57–59]. These methods require for each refinement the resolution of two auxiliary problems, namely one discrete adjoint problem and one discrete error problem which gives ad-

ditional information to the ones provided by the primal problem. On the other hand, the authors in [60] suggested employing a dual weighted residual approach that consists in solving two adjoint problems. Once more, these methods present computational bottlenecks when solving complex high dimensional physical problems.

The method proposed in this work has the advantage of being easily extendable to the case of multi-QoI problems. The extension consists in constructing an error vector for each mesh edge then taking e_k to be the L_2 norm of the error vector. Each component of the latter vector is an error estimation associated with a specific QoI.

3.2.4. Extrapolation technique for higher-dimensional input spaces

A drawback of the direct derivation of the adaptation methodology from the mesh adaptation technique is that, in order to be able to adapt in the whole domain, it is necessary to have nodes (training points) also in each vertex of the hypercube representing the domain when each input is uniformly distributed. However, this can be really limiting when the size of the input variables increases, since the number of vertices of the hypercube rapidly increases as $2^{n_{dim}}$, and subsequently the number of extra training points in which is necessary to evaluate the function of interest augments.

A possible simple solution for this problem, inspired by [26], can be to consider only training points inside the domain and then exploit an extrapolation to cover the remaining part up to the corners and the bounds. The extrapolation procedure can be structured as follows. First, a metamodel is trained on the initial design of experiments, which can be a normal Latin Hypercubes or quasi Monte Carlo design. Then the triangulation of the do-

main is built including also the corners in the set of nodes. It is important to mention that the true function values are not computed at these points. Thus, two different types of edges will be considered: the interpolation edges, constructed by joining two training nodes, and the extrapolation edges, for which at least one of the two nodes is a corner of the domain.

At this point, it is important to highlight the fact that since the actual value of the QoI is not known on the extrapolation edges, one could be less confident about the gradient value computed in the error criteria. Hence, when computing the global error, it is possible to put a smaller weight (or even a null weight) on the gradient part, with respect to the interpolation edges. Then the procedure for adding the new nodes is exactly the same as described in Algorithm 5. While the normal approach is supposed to work at least as fine, or even better, than the extrapolation one for smaller sizes of the input, the latter should behave better when the number of input variables starts to increase, and the number of corners becomes comparable to the size of a DoE to get a sufficiently good metamodel of the QoI.

3.3. Remark on the user-defined parameters

In this section, a brief overview is given on the user-defined parameters of the global algorithmic framework proposed in this work. This is provided because some of them have an important role in the accuracy of the final surrogate, and therefore their value needs to be properly set. In table 1, the user-defined parameters are reported, together with a brief description of their role and the set of values that they can take.

Most of the parameters whose values need to be imposed by the user are related to the sparse PDD algorithm. One of the most relevant is the

Name	Possible values	Role
m	$\mathbb{N}^*$	maximum polynomial degree for PDD
ν	$[1\,..\,d]$	maximum size of ANOVA interaction
p	$[0,1]$	variance threshold for adaptation in sparse-PDD
ϵ	$[0,1]$	error threshold for adaptation in sparse-PDD
Q^2_{tgt}	$[0,1]$	accuracy threshold for adaptation in sparse-PDD
e_{target}	$(0,\infty)$	error threshold in basic refinement algorithm
N_a	$\mathbb{N}^*$	fixed number of added points in refinement algorithm
i_{ref}	$\mathbb{N}$	number of iterations of the adaptation
α	$[0,1]$	error weight in refinement algorithm

Table 1: Parameters

maximum polynomial order m, since it strongly influences the accuracy of the intermediate sparse PDD metamodel, and hence the amount of information added to the final coupled metamodel. As it will be shown in Section 4.1, its value needs to be chosen according to the function of interest, and if necessary a preliminary test can be carried out. A bad value can spoil the convergence of the final metamodel. A possible way to properly set the value of this parameter is by exploiting cross-validation: the surrogate model is trained for different values of m, and the cross validation (CV) errors evaluated for each surrogate, for example the leave-one-out cross validation (LOOCV) error [44, 61]. After evaluating the LOOCV for the surrogates trained at different values of m, it is possible to choose as final value the one allowing for a lower cross validation error. This procedure can be easily implemented, and, in principle, it could be automatized by adding an iteration over different values of m on top of the surrogate training algorithm. Of course, this will increase the total computation cost associated to the training process. In this work, we followed this strategy, with only one difference: since for all the

considered cases we were able to compute the root mean square error (RMSE) [44] estimate for the metamodeling error, we preferred to use this metric for the choice of the parameter instead of the CV estimate. The value of ν, the maximum size of ANOVA interaction terms, is easier to determine, as it can be left equal to the number of variables for smaller input, or fixed to 3 or 4 for higher inputs, following the principle that in most application cases, most relevant interactions occur at lower interaction orders. For the sensitivity of the sparse-PDD to the thresholds p, ϵ and Q^2_{tgt} one can refer to [14]. Author's experience would suggest to fix them to standard values and concentrate mostly on m to improve the convergence of the metamodel. Concerning the adaptation algorithm, the basic algorithm is not recommended in practical applications, since it is very hard to control the number of added points by changing the value of an accuracy threshold e_{target}. Therefore the two main parameters that need to be assigned are the number of added points N_a, the number of iterations i and the error weight α. The normal operational use of the algorithm would be to perform a small number of iterations, even just one, to add a number of nodes according to the available computational resources, if the accuracy of the metamodel trained on the initial experimental design is not satisfactory. A comparison of computations at different values of α is proposed in Section 4.2.

4. Numerical experiments

In this section, several numerical experiments are presented. The objective is to illustrate the performances and the limits of the proposed approach, by making a systematic comparison with classical methods found in litera-

ture.

Two classes of problems are used for validation purposes. First, some well-known algebraic functions, classically used in literature, are tested both without (in Section 4.1) and with mesh adaptation (in Section 4.2). Secondly, the method is tested on two engineering problems in aerospace application (Section 4.3). As it can be observed, the proposed method provides a systematic gain.

In Table 2 the characteristics of the test function used in the work are reported for clarity. Test 1 is a 2D function built with the purpose of testing the whole algorithm. Test 2 and 3 are well-known test functions for metamodels for UQ and optimization taken from literature.

Name	Input dim.	Domain	Function
TEST 1	2	$[-1,1]$	$f(\boldsymbol{x}) = g(10x_1-2)\cos(5x_1^2)\cos(x_2^2)(3-x_2)^2$ with $g(s) = \frac{s\|s\|}{1+s^2}$
TEST 2	3	$[-\pi,\pi]$	$f(\boldsymbol{x}) = \sin x_1 + a\sin^2 x_2 + bx_3^4 \sin x_1$
TEST 3	8	$[0,1]$	$f(\boldsymbol{x}) = \prod_{i=1}^{8} \frac{\|4x_i-2\|+c_i}{1+c_i}$

Table 2: Test functions used for the assessment of the UK-PDD method

4.1. Results without mesh adaptation

The PDD-UK is used in this section to build metamodels for different test functions, verify the convergence of the metamodeling errors with the size of the Experimental Design and compare results with the ones obtained with Ordinary Kriging and sparse-PDD. Furthermore, the sensitivity of the method with respect to some parameters of the sparse adaptive selection of the PDD basis function is analyzed.

Along this section, Latin Hypercube designs of different sizes are used as training points. Each training plan is used to build an Ordinary Kriging metamodel, a sparse-PDD one and a surrogate model exploiting the proposed PDD-UK method. The obtained surrogate models are tested on a LH plan of 100000 points. Comparisons with results obtained also on quasi Monte Carlo (qMC) training sets are also proposed, namely with Halton sampling points.

4.1.1. ***TEST 1****: 2D function*

We introduce the following bivariate function (derived from an univariate test case already used in literature [62]):

$$f(\boldsymbol{x}) = g(10x_1 - 2)\cos(5x_1^2)\cos(x_2^2)(3 - x_2)^2 \quad \text{with} \quad g(s) = \frac{s|s|}{1 + s^2} \tag{33}$$

The function is evaluated in the domain $\boldsymbol{x} \in [-1, 1]^2$. It can be used as first simple case to test the convergence of the proposed method while increasing the size of the experimental design. The test is performed by increasing the number of training points, chosen by means of LH sampling. Note that the RMSE between the metamodels and the true function is computed on 100000 test points. Computations are repeated 15 times for each size of the DoE, in order to account for the variability of the experimental design. Results are reported in Figure 3. It can be seen that the RMSE of the coupled metamodel converges with the increasing of the number of training points and that its mean value is always lower than the ones of the two starting metamodels.

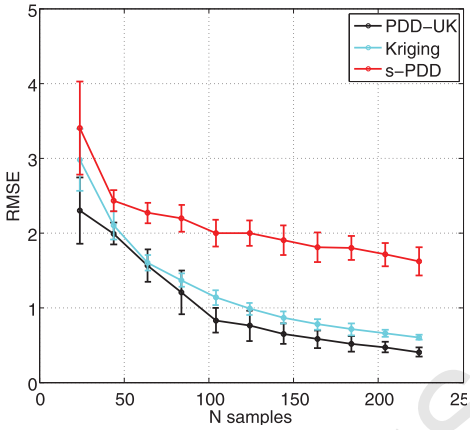


Figure 3: TEST 1: Mean RMSE convergence comparison between Ordinary Kriging, sparse PDD and coupled PDD-UK metamodels, $m = 5$

4.1.2. ***TEST 2****: Ishigami function*

The Ishigami function is another analytical example to verify the convergence of the method and to test the sensibility to some of the parameters. This function, which is widely used for benchmarking in sensitivity analysis, depends on three independent input parameters and can be written as

$$f(\boldsymbol{x}) = \sin x_1 + a \sin^2 x_2 + b x_3^4 \sin x_1 \tag{34}$$

where the input random variables $\boldsymbol{x} = \{x_1, x_2, x_3\}$ are uniformly distributed over $[-\pi, \pi]$. The constants are set to $a = 7$ and $b = 0.1$, as done for example in [12, 14].

A maximum PDD order $m = 3$ is initially considered and the two variance-based (v) and error-based (e) selection algorithms are compared. Table 3 reports values of RMSE, MSE_r and Q^2 (see Appendix A for the definition of

each error estimate) for all the surrogates. It can be noticed from this comparison that the variance-based adaptation approach for the sparse-PDD always produces a less accurate surrogate model with respect to the error based one. For this reason the set of basis functions given to the Universal Kriging in the coupled approach, in the case of variance-based approach, is not enough representative of the function trends for the method to converge, hence no results are obtained. The error based approach, instead, is always able produce a representative set of basis functions and the coupled method is then able to converge. When considering $m = 3$ for the maximum PDD order, OK seems to perform better than sparse-PDD for the Ishigami function. Except for the 40 points training plan, the coupled method always delivers a better surrogate model (in the RMSE and Q^2 sense) than the Ordinary Kriging and the sparse-PDD. This means that the added information in the regression part of the Kriging surrogate is actually able to improve the representation of the function, or, seen from the opposite point of view, that the Kriging departure term is able to improve the representation given only by the sparse-PDD regression. The exception in the 40 points training plan is likely caused by the fact that this set of points is too small to give enough information for the PDD to produce an accurate enough set of basis functions.

The same analysis is repeated when considering a higher maximum PDD order of $m = 10$. In Table 4, results are shown. When using the error-based adaptive approach, sparse-PDD surrogates are better than the Ordinary Kriging ones, and the coupled approach is always able to further reduce the metamodeling error. However, it is important to mention that, as pointed

N_s		OK	s-PDDe	PDD-UKe	s-PDDv	PDD-UKv
40	RMSE	2.19369	3.93583	3.14090	103.9839	-
	MSE_r	0.35438	1.09102	0.70760	1.0050	-
	Q^2	0.413350	0.534050	0.864321	-	-
80	RMSE	1.62157	2.92508	1.04595	74.7111	-
	MSE_r	0.19322	0.62815	8.0481e-2	1.00314	-
	Q^2	0.798325	0.527088	0.940881	-	-
160	RMSE	1.0656	2.53986	0.47179	7.54097	-
	MSE_r	8.27647e-2	0.47480	1.6393e-2	1.12117	-
	Q^2	0.897168	0.519746	0.973951	0.855120e-1	-
320	RMSE	0.66843	2.6088	0.34148	4.09612	-
	MSE_r	3.29050e-2	0.50122	8.58841e-3	0.79921	-
	Q^2	0.959829	0.633514	0.990904	0.572308	-
640	RMSE	0.442623	2.537008	0.227492	3.910519	0.369953
	MSE_r	1.442877e-2	0.474011	3.811484e-3	0.782575	1.007985e-2
	Q^2	0.981296	0.560523	0.995585	0.532891	0.986737

Table 3: TEST 2: Actual error measures for the Ordinary Kriging, Sparse-PDD with error-based (e) and variance-based (v) selection algorithms and coupled PDD-UK surrogate models of the Ishigami function, $m = 3$, $\varepsilon = 10^{-5}$ (variance), $\varepsilon = 10^{-8}$ (error).

out in [14], the computational cost associated with the error-based criterion can be way higher than the one needed to perform the variance-based criterion because a higher number of terms is kept in the sparse representation. While this can be negligible for simple low-dimensional cases such as the Ishigami function, it can be relevant for higher-dimensional problems.

The comparison between the $m = 3$ and the $m = 10$ cases shows, as it could be expected, that this parameter has a high influence also on the quality of the final coupled metamodel, hence it must be chosen wisely. When no information is available about the choice of this parameter, a preliminary convergence study of the sparse-PDD algorithm can be performed at different values of m on the available design of experiments.

For the Ishigami function, the improvement in accuracy given by the

N_s		OK	s-PDDe	PDD-UKe	s-PDDv	PDD-UKv
40	RMSE	2.193697	3.68578	2.19369	-	-
	MSE_r	0.354380	1.00000	0.354380	-	-
	Q^2	0.413350	-	0.413350	-	-
80	RMSE	1.621572	1.08661	1.00083	50.22814	1.625250
	MSE_r	0.19322	8.69541e-2	7.376024e-2	1.30340	0.194121
	Q^2	0.798325	0.994236	0.995617	0.999215	0.792188
160	RMSE	1.06008	0.636108	0.35959	5.358411	1.0424
	MSE_r	8.2764e-2	2.9793e-2	9.5226e-3	2.114611	8.00259e-2
	Q^2	0.897168	0.995448	0.998263	0.999857	0.912285
320	RMSE	0.668438	0.32831	8.42948e-2	5.51245	0.63282
	MSE_r	3.2905e-2	7.93837e-3	5.233157e-4	2.23793	2.94929e-2
	Q^2	0.959829	0.998906	0.999896	0.999981	0.963671
640	RMSE	0.442623	7.291311e-2	1.559595e-2	5.161115	0.321574
	MSE_r	1.442877e-2	3.915368e-4	1.791371e-5	1.961754	7.615962e-3
	Q^2	0.981296	0.999900	0.999994	0.999978	0.989006

Table 4: TEST 2: Actual error measures for the Ordinary Kriging, Sparse-PDD with error-based (e) and variance-based (v) selection algorithms and coupled PDD-UK surrogate models of the Ishigami function, $m = 10$, $\varepsilon = 10^{-5}$ (variance), $\varepsilon = 10^{-8}$ (error).

PDD-UK surrogate with respect to ordinary Kriging and sparse-PDD is tested also on a quasi Monte Carlo (qMC) set of training point, namely on a Halton set [63]. Surrogates are trained on an increasing number of training points, with the same size as in Tables 3 and 4, only for a value of $m = 10$ for the maximum polynomial order. Results for the error-based algorithm for the sparse-PDD are shown in Figure 4. From this plot, the advantage of the PDD-UK in terms of accuracy is clear, especially when increasing the number of training points. Furthermore, the level of accuracy obtained with Halton qMC set is of the same order of magnitude as the one obtained with LHS (see Table 4). Notice that a comparison with results produced with the variance-based selection algorithm was not possible, because it failed to deliver a converged coupled surrogate.

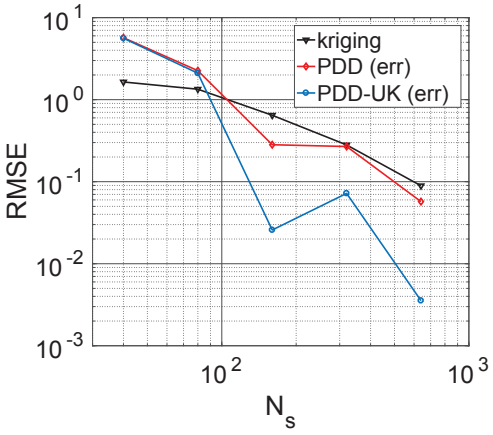


Figure 4: TEST 2: Convergence of the RMSE error measures when increasing the number of training points for Halton qMC training sets. Results are plotted for the Ordinary Kriging, Sparse-PDD and coupled PDD-UK surrogate models of the 3-dimensional Ishigami function at $m = 10$. Error-based adaptive algorithm.

A further test of the convergence of the PDD-UK method can be performed on the sensitivity indices associated to the Ishigami function, since analytical values are known in literature (see for example [14]). A comparison with the numerical values obtained with the three metamodeling techniques under analysis and direct Monte Carlo sampling (as done in [16]) is reported in Table 5. In a practical application the MC approach would be drastically more expensive than the others, due to the high number of model evaluations required (in this particular case we used 100000 Monte Carlo samples), hence the metamodel-based techniques represent a very good trade off between accuracy and efficiency. A general good convergence is shown for the first order sensitivity indices, while the convergence of the second order indices is more difficult for the Monte Carlo based computations, especially when they have

really small (or null) values. This is an intrinsic characteristic of the Monte Carlo computation of the indices, since high order indices computation depends also on the computed values of lower order indices [15], and so it can be spoiled by a loss of accuracy (see for example the negative index in Tab. 5). Hence, the sparse-PDD technique seems to be more suited if the only purpose is the sensitivity analysis, because, due to its close link with ANOVA decomposition, the computation of the indices is straightforward and more accurate, especially for higher order ones. However, as seen before, the UK-PDD metamodel has a lower cross-validation error if the PDD basis is enough representative. Furthermore, another advantage of the PDD-UK metamodel over the sparse-PDD is that, being a Gaussian process metamodel, it can be exploited in several optimization applications, in combination with techniques such as the Expected Improvement and EGO algorithm [23].

SI	Exact	OK	s-PDDv	PDD-UKv	s-PDDe	PDD-UKe	MC
S_1	0.3138	0.3558	0.3133	0.3486	0.3126	0.3137	0.3147
S_2	0.4424	0.4804	0.4397	0.4949	0.4448	0.4431	0.4419
S_3	0	0.0063	0	0.0064	0	0.0046	0.0045
S_{12}	0	0.0053	0	0.0035	0	-0.0002	0
S_{13}	0.2436	0.1462	0.2470	0.1426	0.2423	0.2386	0.2389
S_{23}	0	0.0031	0	0.0022	0	0.0001	0
S_{123}	0	0.0029	0	0.0018	0.0023	0.0001	0
f_0	3.5	3.4861	3.5023	3.4785	3.4983	3.4946	3.4955
D	13.845	11.8388	13.9338	12.7512	13.8550	13.8793	13.8720
Q^2		0.91604	0.99959	0.92383	0.99991	0.99996	
eval.		200	200	200	200	200	100000

Table 5: TEST 2: numerical mean, variance, metamodel accuracy and sensitivity indices of the Ishigami function obtained with different metamodeling techniques and comparison with exact and Monte Carlo results, $m = 10$, $\nu = 3$, $\varepsilon = 10^{-8}$ (error), $\varepsilon = 10^{-3}$ (variance).

4.1.3. ***TEST 3****: 8D Sobol function*

To test the approach on a higher-dimensional problem, the eight-dimensional Sobol function test case is considered [14]. Its expression is:

$$f(\boldsymbol{x}) = \prod_{i=1}^{8} \frac{|4x_i - 2| + c_i}{1 + c_i} \tag{35}$$

where the components of the input vector $\boldsymbol{x}$ are uniformly distributed over $[0, 1]$ and the vector of positive coefficients is $\boldsymbol{c} = \{1, 2, 5, 10, 20, 50, 100, 500\}$.

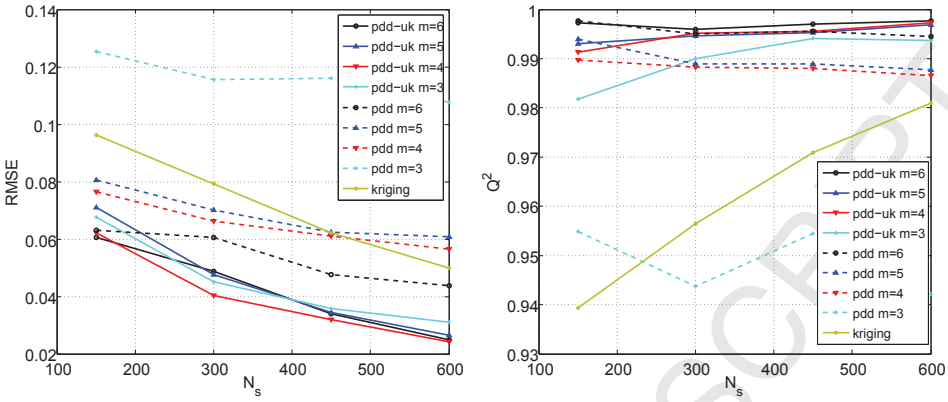
A first comparison between Ordinary Kriging, sparse PDD and the coupled PDD-UK method can be performed on the first order sensitivity indices obtained with the three surrogate modeling techniques. Results are reported in Table 6, and a further comparison is done with analytical (exact) results and values obtained with a classical Monte Carlo sapling performed on the original function. Results show a general good convergence of all methods. It can be noticed that, while being slightly more accurate in some cases, especially for the most influent indices, the coupled PDD-UK approach is not able to outperform the sparse-PDD in the approximation of the sensitivity indices, as observed also for the Ishigami function. Hence, in general, it could be not necessary to perform the MC sampling on the final UK-PDD surrogate to obtain the SIs, but one could simply rely on the use of the PDD coefficients computed during the intermediate construction of the sparse-PDD surrogate. In this way the SIs would come almost effortlessly and with a good accuracy.

A second convergence test can be performed by plotting the trends of the RMSE and Q^2 when increasing the size of the training plan N_s at different

SI	Exact	OK (MC)	s-PDD	PDD-UK (MC)	MC
S_1	0.603	0.654	0.632	0.607	0.603
S_2	0.268	0.265	0.284	0.269	0.271
S_3	0.067	0.045	0.048	0.072	0.069
S_4	0.020	0.015	0.018	0.025	0.022
S_5	0.0055	0.004	0.006	0.010	0.009
S_6	0.000	0.002	0.002	0.005	0.003
S_7	0.000	0.002	0.003	0.004	0.003
S_8	0.000	0.001	0.003	0.004	0.002
f_0	1.0000	1.0064	0.9988	0.9960	0.9998
D	0.1380	0.1185	0.1307	0.1302	0.1378
Q^2		0.9394	0.9896	0.9913	
model evaluations		150	150	150	100000

Table 6: TEST 3: numerical mean, variance, metamodel accuracy and sensitivity indices of the 8-dimensional Sobol function obtained with different metamodeling techniques and comparison with exact and Monte Carlo results, $m = 4$, $\nu = 2$.

values of the maximum polynomial degree m. Values are compared for both the simple sparse-PDD method, the Ordinary Kriging and the couple UK-PDD method. Results are shown in Figure 5. The comparison shows that the convergence of the sparse-PDD is not monotone with the value of m, as already remarked in [14], and this reflects on the convergence of the coupled method. In this context, the more complex and expensive coupling strategy developed in [12] could reduce the sensitivity of the final PDD-UK metamodel to the maximum polynomial order m, but with an increasing computational effort, which could not be justified, as in general all the coupled metamodels at different values of m are better than the single sparse-PDD ones (in the RMSE sense). Another aspect that is important to notice is that the Q^2, and so the cross-validation error, sometimes are not able to capture properly the difference in accuracy of different metamodels, especially when they are



(a) RMSE (b) Q^2

Figure 5: TEST 3: Actual (RMSE) and cross-validation (Q^2) error measures when increasing the number of training points. Results are plotted for the Ordinary Kriging, Sparse-PDD and coupled PDD-UK surrogate models of the 8-dimensional Sobol function at different values of m. Error-based adaptive algorithm.

relatively close. This is a known fact, which must be kept into account in application where is not possible to compute the RMSE.

Also for the 8-dimensional Sobol function, the difference of accuracy of the three metamodeling techniques is compared on Halton qMC training sets of the same size as the ones used in Figure 5. Results are reported in Figure 6. As it can be noticed, even in this case the coupled PDD-UK surrogate is able to improve the accuracy with respect to the two simpler techniques. Notice that the level of accuracy of the surrogate is about the same on the two different kinds of training sets: the RMSE on the LHS set is of the same order of magnitude as the one on the qMC Halton set.

We would like to point out that, for the 8-dimensional Sobol' function, a

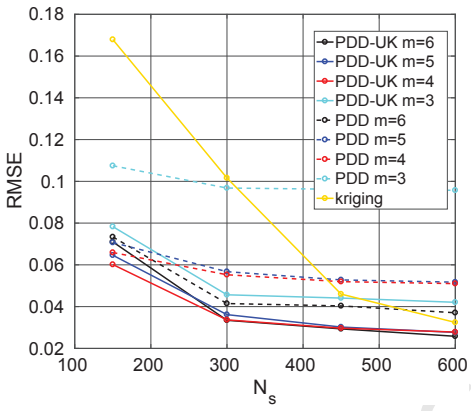


Figure 6: TEST 3: RMSE error measure when increasing the number of training points. Results are plotted for the Ordinary Kriging, Sparse-PDD and coupled PDD-UK surrogate models of the 8-dimensional Sobol function at different values of m. Error-based adaptive algorithm.

comparison between the error-based and the variance-based adaptation criteria was not possible, since, in the several tests performed, the PDD-UK surrogate was not able to converge when using the variance-based selection. This behavior was noticeable, to a certain extent, also on the Ishigami function, and it can be justified by the fact that the sparse basis selected with the variance-based criterion was not accurate enough for the coupled surrogate to converge. Still, as shown in [14], this criterion is very effective for the computation of the sensitivity indices, where it is more efficient than the error-based counterpart.

4.2. Assessment of the adaptive strategy

In this section, the proposed strategy to adaptively add new points to the Experimental Design is assessed. Firstly, the fast approach is compared to the more rigorous one to verify its robustness. Then, several iterations

of the adaptation algorithm are applied to different test functions to test its convergence in RMSE sense. It has to be noticed that in a normal application, just few iterations of the algorithm are likely to be performed, since, if the adequate computational power is available to evaluate the QoI in several training points, it would be more advisable to generate a larger Experimental Design from the beginning.

4.2.1. Comparison between brute and fast approach

First, we present here a comparison between the optimal but computationally expensive approach and the faster one (see Sec. 3.2.3). The analysis is performed on the TEST 1 function (see Table 2).

In Figure 7, results obtained starting from an initial LHS design of 24 points (20 actual LHS points plus the four corners of the bi-dimensional domain) are compared. Comparisons between the brute (a,c,e) and the faster approach (b,d,f) are performed for three different values of α, *i.e.* 0.2, 0.5 and 0.8. Note that curves relative to different values of N_a are shown in different colors. As it is possible to notice, the obtained solutions are, most of the times, practically identical between the two algorithms for both different values of the parameters N_a and α. Sometimes, as it can be seen in Figure 7b for $N_a = 2$, the convergence of the faster algorithm is even better.

Concerning the computational cost of the two algorithms, figure 8 shows the trend of the CPU time when increasing the number of points added at each iteration, *i.e.* N_a, for different values of α. It can be seen that, as expected, the fast algorithm outperforms the brute approach for $N_a > 10$. In fact, the brute approach becomes almost infeasible in this conditions. Note also that, for the fast algorithm, the computational time decreases with

N_a, because less iterations of the adaptation algorithm are required, which includes also the construction of the Delaunay triangulation of the domain. This difference in computational cost is expected to increase with the size of the input space.

The same convergence analysis can be repeated for a larger training plan, this time of 44 training points. The behavior, as shown in Figure 9, is generally the same, with the fast algorithm performing at least as good as the other one.

4.2.2. Convergence

As stated in the previous section, the fast algorithm is able to perform adequately well and allows a bigger flexibility and lower computational cost with respect to the so-called brute approach. Using this assumption, the fast algorithm is then retained for the following analysis. The adaptive part of the algorithm is then tested on TEST 1, 2 and 3 (see Table 2). Note that the convergence is systematically assessed by repeating the adaptation process starting from different LH designs of same size, and comparing then the obtained RMSE mean value and the standard deviation with the ones one would get with standard LHS designs of increasing size. Note also that the number of points added at each iteration is a choice of the user. A sensitivity over this parameter is provided in the following to illustrate the influence on the convergence.

First, we assess the convergence of the algorithm for the TEST 1 function (in this case, fifteen different LH designs of same size are generated). In Figure 10, the RMSE value is represented for each number of training points. it can be seen that, for this test case, the mean error related to the adaptive

strategy converges faster than the one of a simple LHS, especially for $\alpha = 0.5$ and $\alpha = 0.2$, namely when taking into account also the Kriging estimation of the metamodeling error. Furthermore, the adaptation in this particular test case delivers more robust results, since variance of the RMSE is smaller. Note also that a consistent reduction is observed for both cases with 10 and 20 points added for each iteration.

Concerning the TEST 2 function (see Table 2), Figure 11 shows that the adaptive strategy converges faster than the simple LHS, especially for $\alpha = 0.5$ and $\alpha = 0.2$. It also shows that adding a relatively higher number of training points per iteration helps keeping the convergence curve more stable.

We repeat the adaptation process for the TEST 2 function in the case of a Halton qMC initial experimental design, instead of the one from a LHS set. This is done to assess how the adaptation strategy is behaving with a different kind of training points. We note that, since Halton set are composed by deterministic sequences of points, it is not necessary to repeat the process and find an average RMSE. Results are reported in Figure 12. From these results, it is noticeable that, especially for the case with $\alpha = 0.5$, the adapted set of training point is able to produce a better surrogate than a simple Halton qMC set at the first iteration of the adaptation process.

Finally, a test is performed on the TEST 3 function introduced in Table 2, to verify if convergence is retained also on a higher-dimensional case. In this case, the extrapolation method presented in 3.2.4 is assessed. Results for this test-case are reported in Figures 13, 14 and 15. As it can be seen in 13a, the normal approach that puts nodes in the corners of the domain does not show to converge at least as fast as LHS, probably due to the

fact that a relatively high number of training points with respect to the size of the initial DoE (256 over 320) needs to be put in the corners of the domain, leaving too little information to train an adequate metamodel inside the domain. Therefore, the adaptation is repeated with the extrapolating approach in Figure 13b, where all the training points are strictly inside the domain. This is able to improve results and to provide a convergence which is, for the first iteration, much better than the one obtained by just increasing in the size of the LHS plan. This extrapolating approach also allows to reduce the number of training point to a number which is lower than (or very close to) the number of corners, as shown in Figure 14 and 15. It can be however noticed that the convergence of the adaptive approach for this 8-dimensional case tends rapidly to be almost flat when adding 100 new points for each iteration. It is noticeable instead (Figure 14b and 15), when adding 50 samples per iteration, that the metamodeling error decreases generally faster than with the normal LHS: the first iteration decreases strongly the error, while the next iteration tend to realign to LHS, however the trend is better.

Furthermore, for the 8-dimensional TEST 3 case, we repeat the adaptation process starting form Halton qMC initial designs instead of the ones from LHS. Firstly, two iterations of the adaptation algorithm is applied to the Halton training set of 270 points, adding 100 points per iteration, and the RMSE obtained by the PDD-UK surrogate trained on the adapted DoE is compared to the one of a surrogate trained on a Halton set of the same number of points. The same procedure is repeated also with 50 points added per iteration. Results are reported in Figure 16. Then, the same adaptation

procedure is repeated but starting from an initial Halton set of 220 points, adding 50 points per iteration, and the obtained results are shown in Figure 17. In both tested cases, it is possible to see that the adaptation algorithm is effectively working on qMC types of initial DoEs when adding 50 points per iteration, showing a similar behavior as on LHS training sets. This means that the first step of the adaptation process is able to noticeably reduce the RMSE associated to the surrogate model trained on the adapted DoE, while the second iteration is not able to further improve the accuracy. It is possible to notice, for this particular test, a very small dependency of the RMSE with respect to the user-defined parameter α. Instead, when adaptively adding 100 points, the resulting surrogate shows a RMSE that is not improved with respect to a qMC halton set of the same size.

4.3. Engineering Applications

The proposed method is then tested on two different applications in the aerospace field. For these cases, one evaluation of the quantity of interest corresponds to the output of a computer code describing the specific physics of interest.

4.3.1. EXPERT reentry

The engineering application firstly proposed is in the context of the hypersonic flow that occurs during the entry trajectory of the EXPERT space vehicle by the European Space Agency. The goal is the construction of a metamodel for the stagnation pressure p_{st}, which is one of the quantities measured by the sensors flush-mounted in the nose of the vehicle, as function of the freestream values and some chemistry parameters described next [64].

Two points in the entry trajectory of the vehicle are investigated. For each point, nominal freestream conditions are described in table 7. The trajectory point at higher altitudes is known to exhibit chemical non-equilibrium effects in the shock layer, while in the lower point of the trajectory, which is close to the peak heating conditions, the chemistry is mainly in equilibrium.

Altitude, Km	T_∞, K	p_∞, Pa	M_∞
60	245.5	20.3	15.5
30	220	1200	12.3

Table 7: Freestream conditions for the entry trajectory points of the EXPERT vehicle.

The set of equations used to describe the phenomena is a combined physico-chemical model, developed by Barbante [65], able to simulate hypersonic high-temperature reacting flows. Two-dimensional axisymmetric Navier-Stokes equations, supplied with adequate boundary conditions, are combined with the chemical mechanism introduced by Park et al. [66] applied to a mixture of five species air (N, O, NO, N_2 and O_2). Furthermore, the catalyticity of the vehicle surface is taken into account, and it is modeled as a catalytic wall at radiative equilibrium. To simulate the forward problem, we use the in-house code COSMIC developed by Barbante [65]. This solver was designed to approximate hypersonic flow models where chemical non-equilibrium effects need to be accounted for. It includes a hybrid upwind splitting scheme, the hybridization of the van Leer scheme [67] and the Osher scheme [68] and adds a carbuncle fix. An axisymmetric condition is imposed on the symmetry axis, while the wall of the body is modeled by a partially catalytic wall at radiative equilibrium. An example of the solution tempera-

ture field for the nominal conditions at 60km altitude is shown in Figure 18. Each simulation was performed on a single core of an Intel Xeon E5-2680 v3 processor at 2,5 GHz, and resulted in computational times of about one hour per simulation, if restarting from the solution at nominal conditions.

Concerning the uncertainty characterization of the input, the unknown freestream pressure and Mach number p_∞, M_∞ are assumed to follow uninformative uniform distributions, and uniform epistemic uncertainty is associated also to the catalytic recombination coefficient γ, as described in Table 8. Uncertainty is taken into account also on four reactions rates k_r of four chemical dissociation processes, and they are assumed distributed with log-normal distributions (see Table 9).

Variable	Distribution	Minimum	Maximum
p_∞ [Pa] (30km)	Uniform	960	1440
M_∞ (30km)	Uniform	10.9	13.7
p_∞ [Pa] (60km)	Uniform	16.3	24.3
M_∞ (60km)	Uniform	13.7	17.3
γ	Uniform	0.001	0.002

Table 8: Uncertainties on freestream conditions and catalytic recombination constant.

Gas reaction	Distribution of $\log_{10} k_r$	σ_r
$NO + O \rightarrow N + O + O$	Normal	0.12
$NO + N \rightarrow N + O + N$	Normal	0.12
$O_2 + N_2 \rightarrow 2O + N_2$	Normal	0.10
$O_2 + O \rightarrow 2O + O$	Normal	0.10

Table 9: Uncertainties on gas reaction rates.

In Tables 10 and 11, RMSEs are compared to assess the quality of the

different surrogate models for two different points of the trajectory. Three different Latin Hypercubes designs of experiments of increasing size (respectively 120, 680 and 3060 points) are utilized, to state the convergence of the combined method and to verify the gain in efficiency with respect to the other two techniques.

N_s	OK	s-PDD	PDD-UK
120	1452	1252	423
680	388	767	286
3060	364	726	210

Table 10: EXPERT reentry, 30km point, RMSE comparison of the metamodels for the stagnation pressure, $\nu = 2$, $m = 4$ for the 120 plan, $m = 5$ for the 680 and $m = 6$ for the 3060, $\varepsilon_{Q^2} = 10^{-7}$.

N_s	OK	s-PDD	PDD-UK
120	50.577	23.563	19.945
680	14.918	20.458	13.875
3060	12.196	16.565	10.258

Table 11: EXPERT reentry, 60km point, RMSE comparison of the surrogate models for the stagnation pressure, $\nu = 2$, $m = 4$ for the 120 and 680 plan,$m = 6$ for the 3060 $\varepsilon_{Q^2} = 10^{-7}$.

It can be seen that, for both trajectory points, the metamodeling error associated to the stagnation pressure is lower for the PDD-UK surrogate. This happens because the PDD method is well converging and so the selected basis function are able to add useful information to the Universal Kriging. The weak point in the training algorithm appears to be good choice of the s-PDD parameters and in general the convergence of this part of the metamodel. When a good PDD metamodel is trained, often the coupled method has a

lower RMSE than both simple techniques, otherwise it can be worse than any of them.

4.3.2. TACOT ablation test case

A higher dimensional engineering case is proposed in here with the analysis of the temperature of an ablative material at a fixed position and imposed time of an ablation process. In particular, let us consider the unidirectional ablation of a 7.21cm thick TACOT (Theoretical Ablative Composite for Open Testing) material, exposed to a constant heat flux for one minute before radiatively cooling down. This rectangular incoming flux, is an interesting case to test the method proposed in this paper. While this case does not represent the atmospheric entry of a spacecraft, it is however quite close to an ablation test in the Plasmatron facility [69].

Uncertainties are considered on 27 input parameters related to the physical and chemical properties of the material. A uniform distribution is associated to each uncertain variable, with values reported in Table 12. The quantity of interest is the temperature of the material at a position of $x = 5.61$cm, meaning 1.6cm from the heated surface, at the time $t = 80$s over 120s of simulation (see Figure 19), performed with the PATO code [70]. Each simulation is performed on one core of an Intel Xeon E5-2680 v3 processor at 2,5 GHz, and this results in simulation times of about ten minutes for each configuration. In order to reduce the computational effort while not affecting the metamodeling accuracy, the PDD-UK is built just on the input variables which are not completely neglected in the sparse-PDD regression. This leads to considering only 18 input dimensions in the final metamodel.

The comparison between the three metamodeling techniques is reported

in Table 13. First of all, it can be noticed that each one of the three techniques shows to converge when increasing the size of the Latin Hypercubes Experimental Design. For the smaller design with size of 200 samples, the best performing surrogate is Ordinary Kriging. Since more than 100 basis functions are kept in sparse PDD representation with $\nu = 2$, $m = 4$, the computation of the coefficients and so the choice of the rejected basis function can not be accurate enough for the smaller designs, hence also the coupled PDD-UK metamodel suffers the inaccurate choice of regression functions, and results less accurate than Ordinary Kriging. However, when enough training points are considered, the metamodeling error of the PDD-UK becomes the smaller of the three techniques. The convergence of the method for smaller DoE could be improved by reducing the number of terms kept in the final regression by reducing the maximum polynomial order and the order of interaction, as shown in the case 2, with $\nu = 1$, $m = 2$. The RMSE values obtained for optimized values of ν and m parameters is reported in table 14. The maximum ANOVA interaction is set to $\nu = 2$, and m is kept equal to one for the smaller training plans and then increased to two at 700 training points. Note the consistent gain in accuracy obtained with the coupled metamodel for smaller experimental design, while for the bigger one the error seems to be at convergence. Note that the choice of $\nu = 2$ and m can be easily automatized by picking the best combination in terms of leave-one-out cross validation error.

5. Conclusions

This paper proposed an algorithm for the construction of efficient surrogate models for cases when expensive numerical simulations are required to evaluate the functions of interest.

First, an optimal basis is constructed, consisting in the use of sparse polynomials selected by sparse Polynomial Dimensional Decomposition as basis functions for the regression term of Universal Kriging surrogate model. This improves the convergence of the metamodel with respect to both Ordinary Kriging and Polynomial Dimensional Decomposition, as tested on different test cases. Secondly, the algorithm is supplemented with a method to adaptively enrich the set of training points. It relies on the construction of a n-dimensional Delaunay grid in the space of the input variables, using the sampling points as nodes of the grid, and then in the application of an adaptation algorithm derived from mesh adaptation techniques. It consist in adding point to minimize an error measure that balances the estimate of the metamodeling error with the function gradients. Moreover, the algorithm allows the choice of a specific number of points to add, which be could very useful when dealing with computational cost constraints. The convergence of this technique has been tested on different test functions with input spaces up to 8-dimensions. The algorithm shows to perform systematically better than the simple use of a more refined Latin Hypercube sample, especially for the lower dimensional test cases and when considering just one adaptation iteration to add a few points to an already meaningful experimental design.

Finally, the proposed algorithm has been tested on two medium-to-high-dimensional engineering applications in the aerospace context. The first one,

taking into account 7 uncertain inputs, consisted in building a surrogate model for the stagnation pressure of the hypersonic entry flow around the node of the EXPERT vehicle. While the second one involved the construction of a surrogate for predicting the temperature in a fixed point of an ablating material, as a function of 27 input parameters. In both cases, very good results in terms of accuracy and computational cost have been obtained with respect to the classical sparse-PDD construction, therefore highlighting the interest of the proposed approach.

Appendix A. Metamodel assessment

Several error measures can be found in literature to state the quality of the metamodel, that is its local or global difference with respect to the actual function of interest. In this paper, different techniques are used, for the sake of comparison with previous works. In here, the reader can find a definition of the used error measures.

If one can afford to compute the actual value of the quantity of interest in different points of the stochastic space other than the training points, it is easy then to compute the difference at those points between the actual solution and the prediction given by the surrogate. Then it is possible to integrate this local error to obtain the actual global root mean squared error (RMSE). In practice the integration is done by a numerical integration technique [44]:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}\left(f(\boldsymbol{x}_i) - \hat{f}(\boldsymbol{x}_i)\right)^2} \tag{A.1}$$

The same information can be used to compute a normalized measure called

relative mean square error MSE_r [12]:

$$\mathrm{MSE}_r = \frac{\sum_{i=1}^{N_t}\big(f(\boldsymbol{x}_i) - \hat{f}(\boldsymbol{x}_i)\big)^2}{\sum_{i=1}^{N_t}\big(f(\boldsymbol{x}_i) - \hat{\mu}_y\big)^2} \tag{A.2}$$

where $\hat{\mu}_y$ is the estimated mean of the output variable.

Often, in practical applications the computational cost associated to the evaluation of the model solution is too high, hence the RMSE can not be directly computed. In these cases an estimate of the error measure is instead required. One of the most common global error estimates in the literature is *leave-one-out cross validation* (CV) [44, 61]. It consists in fitting the Kriging surrogate model on $N_s - 1$ points, by leaving out one training point at a time, then the response is predicted at this point with the metamodel. Then the CV error can be defined as

$$\mathrm{CV} = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}\Big(f_i - \hat{f}_i^{(-i)}\Big)^2} \tag{A.3}$$

where f_i is the training point observed response, while $\hat{f}_i^{(-i)}$ is the prediction at the left-out point using the surrogate built from all the other points. An easier interpretation of the CV error can be achieved by computing the determination coefficient Q^2

$$\mathrm{Q}^2 = 1 - \frac{\mathrm{CV}^2}{\hat{V}[\mathcal{Y}]} \tag{A.4}$$

where

$$\hat{V}[\mathcal{Y}] = \frac{1}{N_s - 1}\sum_{i=1}^{N_s}(y_i - \bar{y})^2 \qquad \text{with} \quad \bar{y} = \frac{1}{N_s}\sum_{i=1}^{N_s} y_i \tag{A.5}$$

Hence if Q^2 is close to unity it means that the metamodel is able to well fit the function of interest.

References

[1] D.R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345 – 383, 2001.

[2] O. P. Le Maître and O. M. Knio. *Spectral methods for uncertainty quantification*. Springer, 2010.

[3] D. Xiu and G. E. Karniadakis. The wiener-askey polynomial chaos for stochastic differential equations. *SIAM J Sci Comput*, 24(2):619 – 644, 2002.

[4] S. Rahman. A polynomial dimensional decomposition for stochastic computing. *International Journal for Numerical Methods in Engineering*, 76:2191 – 2116, 2008.

[5] M. Buhmann. *Radial Basis Functions*. Cambridge University Press, 2003.

[6] N. A. C. Cressie. *Statistics for spatial data*. John Wiley & Sons, 1993.

[7] Enrica Bernardini, Seymour M.J. Spence, Daniel Wei, and Ahsan Kareem. Aerodynamic shape optimization of civil structures: A CFD-enabled Kriging-based approach. *Journal of Wind Engineering and Industrial Aerodynamics*, 144:154 – 164, 2015.

[8] V. Dubourg and B. Sudret. Meta-model-based importance sampling for reliability sensitivity analysis. *Structural Safety*, 49:27 – 36, 2014.

[9] B. Echard, N. Gayton, M. Lemaire, and N. Relun. A combined Importance Sampling and Kriging reliability method for small failure probabilities with time-demanding numerical models. *Reliability Engineering and System Safety*, 111:232 – 240, 2012.

[10] Jouke de Baar, Stephen Roberts, Richard Dwight, and Benoit Mallol. Uncertainty quantification for a sailing yacht hull, using multi-fidelity kriging. *Computers and Fluids*, 123:185 – 201, 2015.

[11] Pierre Barbillon, Gilles Celeux, Agnès Grimaud, Yannick Lefebvre, and Étienne De Rocquigny. Nonlinear methods for inverse statistical problems. *Computational Statistics and Data Analysis*, 55:132–142, 2011.

[12] P. Kersaudy, B. Sudret, N. Varsier, O. Picon, and J. Wiart. A new surrogate modeling technique combining Kriging and polynomial chaos expansions - Application to uncertain analysis in computational dosimetry. *Journal of Computational Physics*, 286:103 – 117, 2015.

[13] V. Roshan Joseph, Ying Hung, and Agus Sudjianto. Blind Kriging: A new method for developing metamodels. *ASME Journal of Mechanical Design*, (130), 2008.

[14] K. Tang, P. M. Congedo, and R. Abgrall. Adaptive surrogate modeling by ANOVA and sparse polynomial dimensional decomposition for global sensitivity analysis in fluid simulation. *Journal of Computational Physics*, 314:557–589, 2016.

[15] I. M. Sobol'. Sensitivity estimates for nonlinear mathematical models. *Mathematical modelling & Computational Experiments*, 1:407–414, 1993.

[16] I. M. Sobol'. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulations*, 55:271–280, 2001.

[17] V. Yadav and S. Rahman. Adaptive-sparse polynomial dimensional decomposition methods for high-dimensional stochastic computing. *Comput. Methods Appl. Mech. Engrg.*, 274:56 – 83, 2014.

[18] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239 – 245, 1979.

[19] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard. DACE: a MATLAB kriging toolbox, version 2.0. Technical report, Technical University of Denmark, 2002.

[20] J. S. Park. Optimal latin-hypercube designs for computer experiments. *J. Stat. Plan. Inference*, 39:95 – 111, 1994.

[21] G. G. Wang. Adaptive response surface method using inherited latin hypercube design points. *Journal of Mechanical Design*, 125:210 – 220, 2003.

[22] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical sciences*, 4(4):409 – 435, 1989.

[23] D.R Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455 – 492, 1998.

[24] J. A.S. Witteveen, A. Loeven, and H. Bijl. An adaptive stochastic finite elements approach based on newton-cotes quadrature in simplex elements. *Computers & Fluids*, 38:1270 – 1288, 2009.

[25] J. A.S. Witteveen and G. Iaccarino. Refinement criteria for simplex stochastic collocation with local extremum diminishing robustness. *SIAM J. Sci. Comput.*, 34(3):A1522 – A1543, 2012.

[26] J. A.S. Witteveen and G. Iaccarino. Simplex stochastic collocation with random sampling and extrapolation for nonhypercube probability spaces. *SIAM J. Sci. Comput.*, 34(2):A814 – A838, 2012.

[27] J. A.S. Witteveen and G. Iaccarino. Simplex stochastic collocation with ENO-type stencil selection for robust uncertainty quantification. *Journal of Computational Physics*, 239:1 – 21, 2013.

[28] T. Coupez. Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing. *Journal of Computational Physics*, 230:2391 – 2405, 2011.

[29] T. Coupez, G. Jannoun, N. Nassif, H.C. Nguyen, H. Digonnet, and E. Hachem. Adaptive time-step with anisotropic meshing for incompressible flows. *Journal of Computational Physics*, 241:195 – 211, 2013.

[30] Charles Audet and Jr. J. E. Dennis. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.

[31] A. Conn, K. Scheinberg, and L. Vicente. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, 2009.

[32] H.-M. Gutmann. A radial basis function method for global optimization. *Journal of Global Optimization*, 19(3):201–227, 2001.

[33] Rommel G. Regis and Christine A. Shoemaker. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing*, 19(4):497–509, 2007.

[34] Rommel G. Regis. Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. *Comput. Oper. Res.*, 38(5):837–853, May 2011.

[35] Stefan Jakobsson, Michael Patriksson, Johan Rudholm, and Adam Wojciechowski. A method for simulation based optimization using radial basis functions. *Optimization and Engineering*, 11(4):501–532, 2010.

[36] Géraud Blatman and Bruno Sudret. An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, 25(2):183 – 197, 2010.

[37] Géraud Blatman and Bruno Sudret. Efficient computation of global sensitivity indices using sparse polynomial chaos expansions. *Reliability Engineering & System Safety*, 95(11):1216 – 1229, 2010.

[38] Xiu Yang, Minseok Choi, Guang Lin, and George Em Karniadakis. Adaptive anova decomposition of stochastic incompressible and compressible flows. *Journal of Computational Physics*, 231(4):1587 – 1614, 2012.

[39] Zhongqiang Zhang, Minseok Choi, and George Em Karniadakis. Anchor points matter in anova decomposition. In Jan S. Hesthaven and Einar M. Rønquist, editors, *Spectral and High Order Methods for Partial Differential Equations*, pages 347–355, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[40] Xiang Ma and Nicholas Zabaras. An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations. *Journal of Computational Physics*, 229(10):3884 – 3915, 2010.

[41] R. Caflisch, W. Morokoff, and A. Owen. Valuation of mortgage-backed securities using brownian bridges to reduce the effective dimension. *Journal of Computational Finance*, 1:27–46, 1997.

[42] G. Matheron. *The theory of regionalised variables and its applications*. PhD thesis, École Nationale Supérieure des Mines, 1971.

[43] V. Picheny. *Improving accuracy and compensating for uncertainty in surrogate modeling*. PhD thesis, École Nationale Supérieure des Mines, 2009.

[44] T. Goel, R. T. Hafkta, and W. Shyy. Comparing error estimation mea-

sures for polynomial and kriging approximation of noise-free functions. *Structural and multidisciplinary optimization*, 38:429 – 442, 2009.

[45] P. G. Constantine, E. Dow, and Q. Wang. Active subspace method in theory and practice: application to kriging surfaces. *SIAM J. Sci. Comput.*, 36(4):A1500 – A1524, 2014.

[46] Luca Margheri and Pierre Sagaut. A hybrid anchored-ANOVA – POD/kriging method for uncertainty quantification in unsteady high-fidelity CFD simulations. *Journal of Computational Physics*, 324:137 – 173, 2016.

[47] R.M. Neal. Monte carlo implementation of gaussian process models for bayesian regression and classification. Technical Report Tech. Rep. 9702, Department of statistics, University of Toronto, Toronto, Ontario, Canada, 1997.

[48] J. Sacks, S. B. Schiller, and Welch W. J. Designs for computer experiments. *Technometrics*, 31(1):41–47, 1989.

[49] G Jannoun. *Space-Time accurate anisotropic adaptation and stabilized finite element methods for the resolution of unsteady CFD problems*. Theses, Ecole Nationale Supérieure des Mines de Paris, 2014.

[50] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math*, 4:237–264, 1996.

[51] P. Houston and Endre Süli. Stabilised hp-finite element approximation

of partial differential equations with nonnegative characteristic form. *Computing*, 66(2):99–119, 2001.

[52] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2001.

[53] M. B. Giles and E. Süli. Adjoint methods for pdes: a posteriori error analysis and postprocessing by duality. *Acta Numerica*, 11:145–236, 2002.

[54] L. Demkowicz, J. Gopalakrishnan, and A. H. Niemi. A class of discontinuous petrov–galerkin methods. part iii: Adaptivity. *Applied Numerical Mathematics*, 62(4):396 – 427, 2012.

[55] A. T. Patera and Jaume Peraire. *A General Lagrangian Formulation for the Computation of A Posteriori Finite Element Bounds*, pages 159–206. Springer Berlin Heidelberg, 2003.

[56] S. I. Repin. A posteriori error estimates for optimal control problems. *IFAC Proceedings Volumes*, 42(2):85 – 90, 2009. 14th IFAC Workshop on Control Applications of Optimization.

[57] R. Hartmann. Multitarget error estimation and adaptivity in aerodynamic flow simulations. *SIAM Journal on Scientific Computing*, 31(1):708–731, 2008.

[58] K. Kenan, S. Prudhomme, L. Chamoin, and M. Laforest. A new goal-oriented formulation of the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 327:256–276, 2017.

[59] E.H. van Brummelen, S. Zhuk, and G.J. van Zwieten. Worst-case multi-objective error estimation and adaptivity. *Computer Methods in Applied Mechanics and Engineering*, 313:723 – 743, 2017.

[60] B. Endtmayer and T. Wick. A partition-of-unity dual-weighted residual approach for multi-objective goal functional error estimation applied to elliptic problems. 17, 02 2017.

[61] M. Meckesheimer, R. R. Barton, T. W. Simpson, and A. Booker. Computationally inexpensive metamodel assessment strategies. *AIAA Journal*, 40(10):2053 – 2060, 2002.

[62] Abdellah Chkifa, Albert Cohen, Pierre-Yves Passaggia, and Jacques Peter. A comparative study between kriging and adaptive sparse tensor-product methods for multi-dimensional approximation problems in aerodynamics design. *ESAIM: PROCEEDINGS AND SURVEYS*, 48:248–261, 2015.

[63] J.H. HALTON. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960.

[64] J. Tryoen, P. M. Congedo, R. Abgrall, N. Villedieu, and T.E. Magin. Bayesian-based method with metamodels for rebuilding freestream conditions in atmospheric entry flows. *AIAA Journal*, 52(10):2190 – 2197, 2014.

[65] P. Barbante. *Accurate and efficient modelling of high temperature non-equilibrium air flows*. PhD thesis, Von Karman Institute, 2001.

[66] C. Park, R. Jaffe, and H. Partridge. Chemical-kinetic parameters of hyperbolic earth entry. *Journal of Thermophysics and Heat Transfer*, 15(1):76 – 90, 2001.

[67] B. Van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to godunov's method. *Journal of Computational Physics*, 32(1):101–136, 1979.

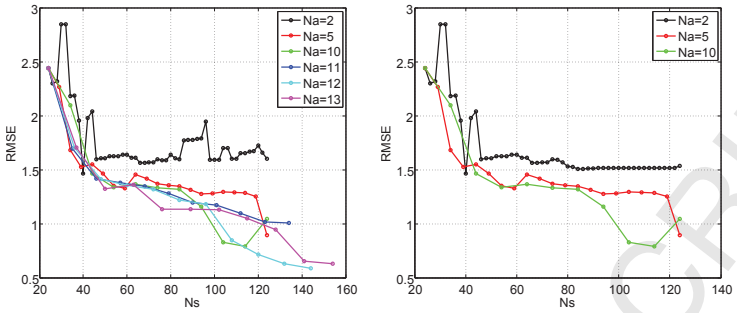
[68] S. Osher and F. Solomon. Upwind Difference Schemes for Hyperbolic Systems of Conservation Laws. *Mathematics of Computation*, 38(158):339–374, 1982.

[69] B. Bottin, O. Chazot, M. Carbonaro, V. Van Der Haegen, and S. Paris. *The VKI Plasmatron Characteristics and Performance*. Defense Technical Information Center, 2000.

[70] Jean Lachaud and Nagi N. Mansour. Porous-material analysis toolbox based on openfoam and applications. *Journal of Thermophysics and Heat Transfer*, 28(2):191–202, 2014.

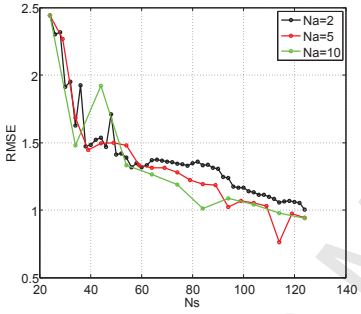
Algorithm 5 Fixed N_a adaptation, fast approach

Calculate e_k for each $k = 1, \ldots, n_e$
Compute $\lambda = (\sum_{k=1}^{n_e} \frac{e_k^{1/3}}{A})^3$, where $A = N + N_a$ and $s_k = (\frac{\lambda}{e_k})^{1/3}$
Compute $N_k = \lfloor \frac{1}{s_k} \rfloor$
while $k \leq n_{edges}$ and $\sum_k n_p < N_a$ **do**
  if $(N_k > 0)$ **then**
    for $j = 1 : N_k$ **do**
      $n_{p_k} = n_{p_k} + 1$
      if $\sum_i n_p = N_a$ **then**
        EXIT
      end if
    end for
  end if
  $i = i + 1$
end while
if $\sum_i n_p < N_a$ **then**
  Compute $s_k^{new} = \frac{1}{s_k} - N_k$
  Sort the edges k according to their value of s_k^{new} in decreasing order and compute $N_k^{new} = \lfloor \frac{1}{s_k^{new}} \rfloor + 1$
  while $k \leq n_{edges}$ and $\sum_k n_p < N_a$ **do**
    if $(N_k^{new} > 0)$ **then**
      for $j = 1 : N_k + 1$ **do**
        $n_{p_k} = n_{p_k} + 1$
        if $\sum_i n_p = N_a$ **then**
          EXIT
        end if
      end for
    end if
    $k = k + 1$
  end while
end if
for each edge k where $n_{p_k} > 0$ **do**
  add n_{p_k} evenly spaced new points along the edge
end for

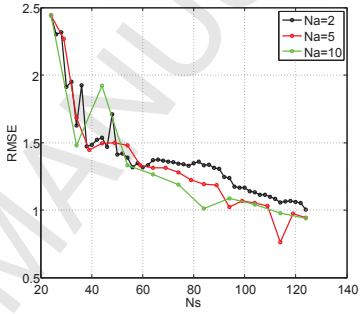


(a) brute, $\alpha = 0.8$

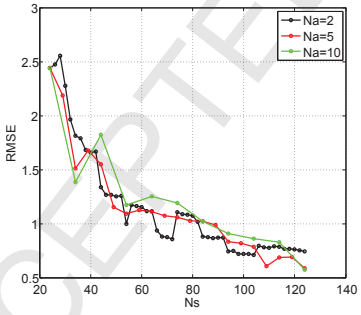
(b) fast, $\alpha = 0.8$



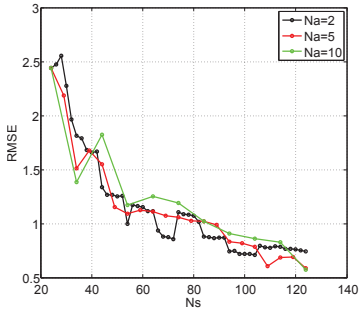
(c) brute, $\alpha = 0.5$



(d) fast, $\alpha = 0.5$

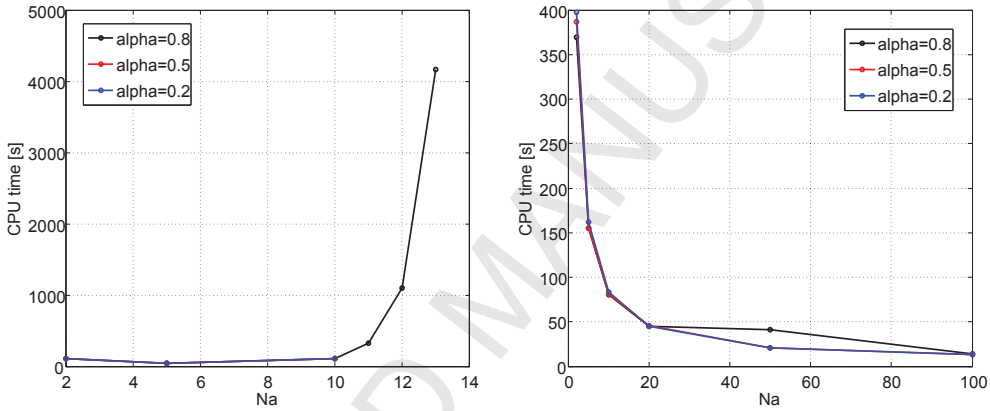


(e) brute, $\alpha = 0.2$



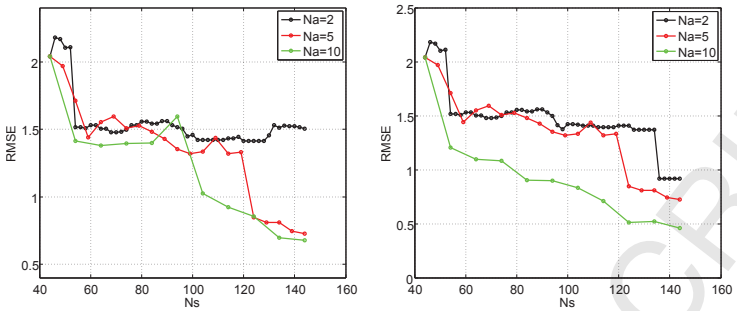
(f) fast, $\alpha = 0.2$

Figure 7: TEST 1: Comparison between brute and fast approach, initial DoE of 24 points



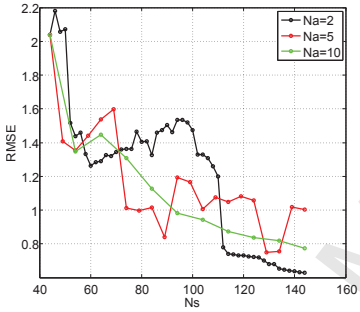
(a) brute (b) fast

Figure 8: TEST 1: Comparison between the computational cost of brute and fast approach, initial DoE of 24 points

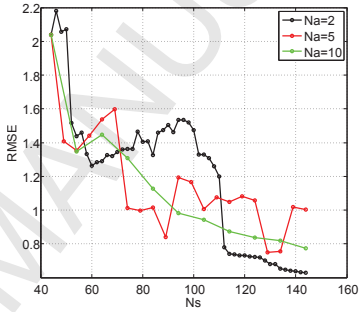


(a) brute, $\alpha = 0.8$

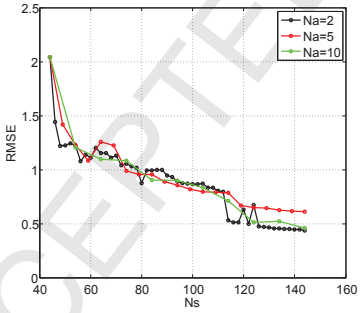
(b) fast, $\alpha = 0.8$



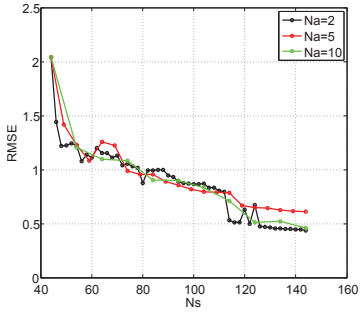
(c) brute, $\alpha = 0.5$



(d) fast, $\alpha = 0.5$

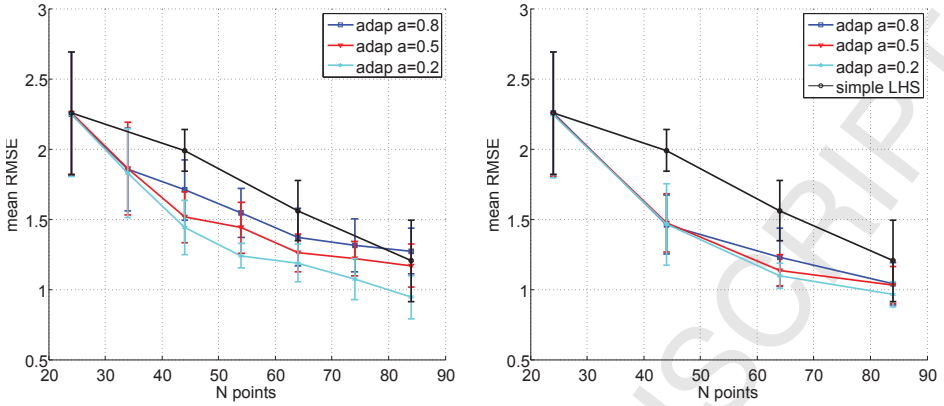


(e) brute, $\alpha = 0.2$



(f) fast, $\alpha = 0.2$

Figure 9: TEST 1: Comparison between brute and fast approach, initial LHS of 44 points



(a) 10 points per iteration

(b) 20 points per iteration

Figure 10: TEST 1: Convergence of the mean value of the RMSE and corresponding deviation computed with 15 different starting LHS DoE. The result of a simple increase of LHS point is compared with adaptation at different values of α coefficient

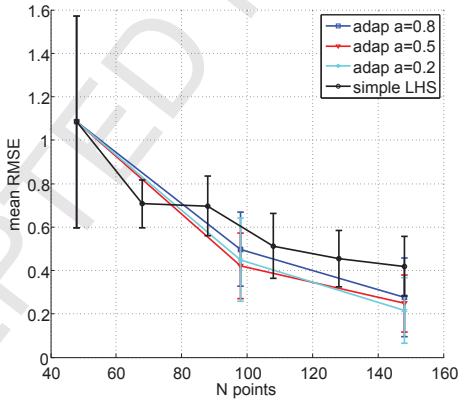


Figure 11: TEST 2: Convergence of the mean value of the RMSE and corresponding deviation computed with 15 different starting LHS DoE. The result of a simple increase of the LHS is compared with adaptation at different values of α coefficient

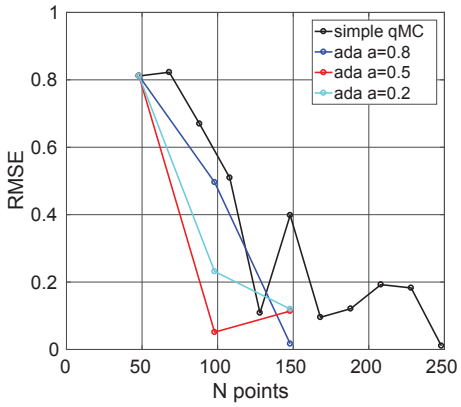
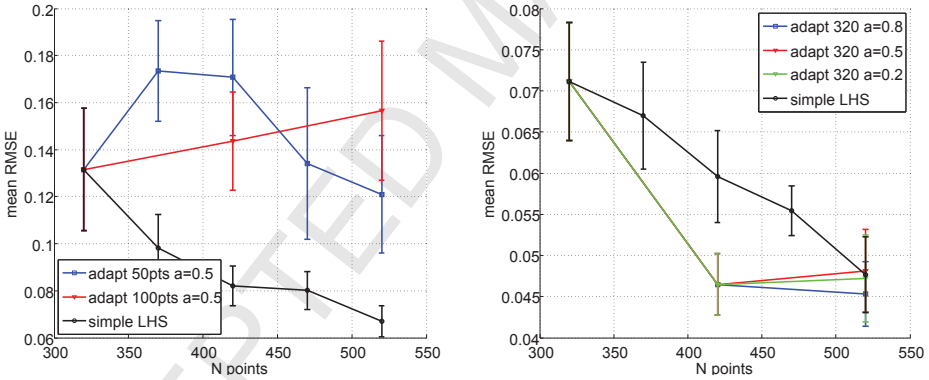
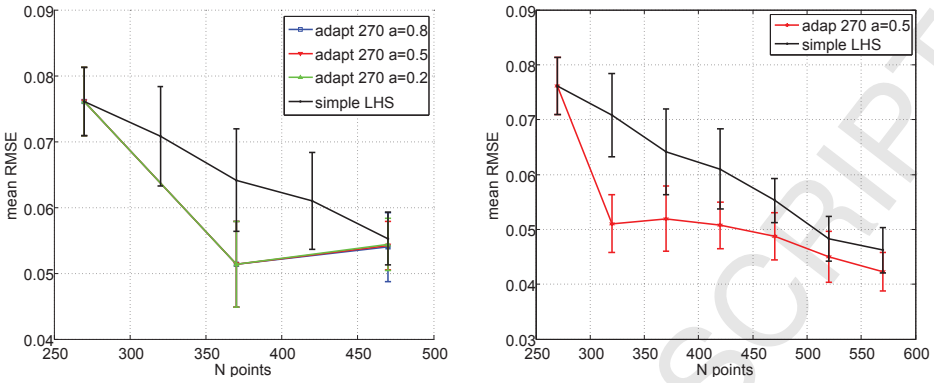


Figure 12: TEST 2: Convergence of the RMSE. The result of a simple increase of the qMC Halton set is compared with adaptation at different values of α coefficient



(a) without extrapolation (b) with extrapolation

Figure 13: TEST 3: Convergence of the mean value of the RMSE and corresponding deviation computed with 14 different starting LHS DoE of 320 points. The result of a simple increase of LHS points is compared with adaptation at different values of α coefficient. A comparison is done between the normal algorithm (a) and the one with extrapolation near the corners of the domain (b).



(a) 100 (b) 50

Figure 14: TEST 3: Convergence of the mean value of the RMSE and corresponding deviation computed with 7 different starting LHS DoE of 270 points. The result of a simple increase of LHS points is compared with adaptation at different values of α coefficient. A comparison is done between the addition of 100 and 50 points per iteration.

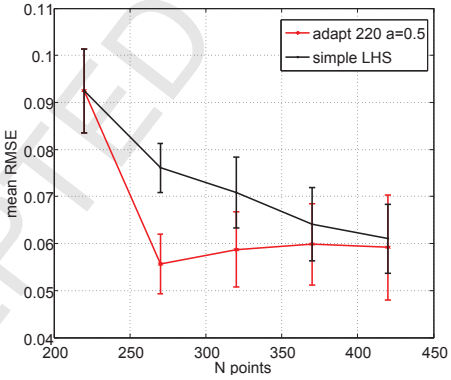
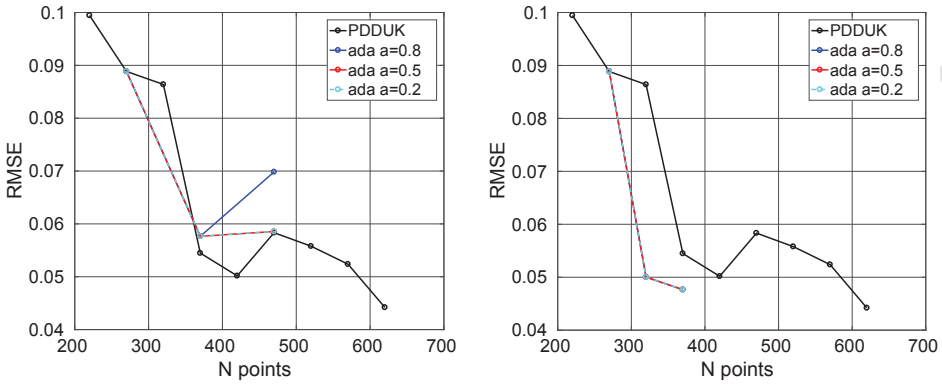


Figure 15: TEST 3: Convergence of the mean value of the RMSE and corresponding deviation computed with 7 different starting LHS DoE of 220 points. The result of a simple increase of LHS points is compared with adaptation at different values of α coefficient.



(a) without extrapolation (b) with extrapolation

Figure 16: TEST 3: Convergence of the RMSE computed on a starting Halton set of 270 points. The result of a simple increase of qMC Halton points is compared with adaptation at different values of α coefficient. results are reported for the addition of 100 (a) and 50 (b) points per iteration of the adaptive algorithm.

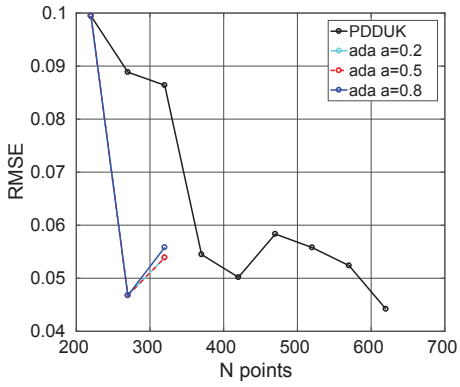


Figure 17: TEST 3: Convergence of the RMSE computed on a starting Halton set of 220 points. The result of a simple increase of qMC points is compared with adaptation at different values of α coefficient.

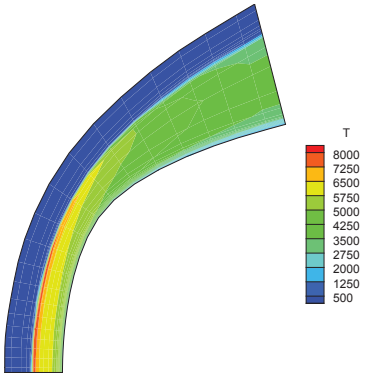


Figure 18: Simulation of the hypersonic flow around the nose of EXPERT vehicle, performed with COSMIC code: temperature filed at nominal freestream conditions at 60km altitude.

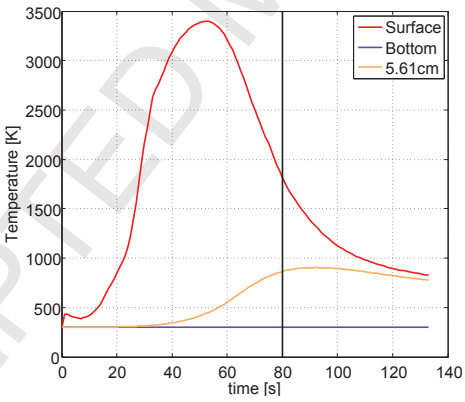


Figure 19: Temperature trend of the reference point at 5.61cm inside TACOT material compared to the one of the heated surface, obtained with nominal material parameters. The black vertical line indicates the reference time at which the sensitivity analysis is carried out.

Variable	Description	Minimum	Maximum
ρ_f	Fiber density	1520	1680
ϵ_f	Fiber volume fraction	0.095	0.105
ρ_m	Matrix density	1140	1260
ϵ_m	Matrix volume fraction	0.095	0.105
K_v	Permeability of the virgin material	1.52e-11	1.68e-11
K_c	Permeability of the char	1.9e-11	2.1e-11
C	Carbon fraction	0.1854	0.2266
H	Hydrogen fraction	0.6111	0.7469
O	Oxygen fraction	0.1035	0.1265
A_1	Pre-exponential factor reaction 1	10800	13200
e_1	Activation energy reaction 1	64017.801	78243.979
h_1	Pyrolysis enthalpy reaction 1	-4.4e6	-3.6e6
A_2	Pre-exponential factor reaction 2	4.479993e8	5.475547e8
e_2	Activation energy reaction 2	1.529775e5	1.869725e5
h_2	Pyrolysis enthalpy reaction 2	-4.4e6	-3.6e6
c_p	Heat capacity virgin	0.95	1.05
k_i	Conductivity i virgin	0.95	1.05
k_j	Conductivity j virgin	0.95	1.05
k_k	Conductivity k virgin	0.95	1.05
C_e	Emissivity virgin	0.95	1.05
C_r	Reflectivity virgin	0.95	1.05
c_p	Heat capacity char	0.95	1.05
k_i	Conductivity i char	0.95	1.05
k_j	Conductivity j char	0.95	1.05
k_k	Conductivity k char	0.95	1.05
C_e	Emissivity char	0.95	1.05
C_r	Reflectivity char	0.95	1.05

Table 12: Uncertainties characterization for PATO: minimum and maximum of the uniform distribution associated to each uncertain input.

N_s	OK	s-PDD 1	PDD-UK 1	s-PDD 2	PDD-UK 2
200	2.4191	3.8836	3.8798	2.9348	2.5952
300	1.5707	3.3070	2.7257	2.6766	1.4420
400	1.2439	3.0733	1.8667	2.6449	1.1829
500	0.9394	1.9956	0.6890	2.6574	0.8736
600	0.7055	1.6047	0.4821	2.5490	0.6620
700	0.5779	1.3625	0.4470	2.5658	0.5427

Table 13: PATO, RMSE comparison of the metamodels, $\nu = 2$, $m = 4$ for case 1, $\nu = 1$, $m = 2$ for case 2.

N_s	OK	s-PDD	PDD-UK
200	2.4191	2.4103	1.6193
300	1.5707	2.2190	0.8193
400	1.2439	1.8097	0.5365
500	0.9394	1.4698	0.4777
600	0.7055	1.3344	0.4235
700	0.5779	1.1882	0.4232

Table 14: PATO, RMSE comparison of the ordinary Kriging, sparse-PDD and PDD-UK metamodels, with optimized parameter m for the PDD at each training set. A value of $\nu = 2$ have been used.